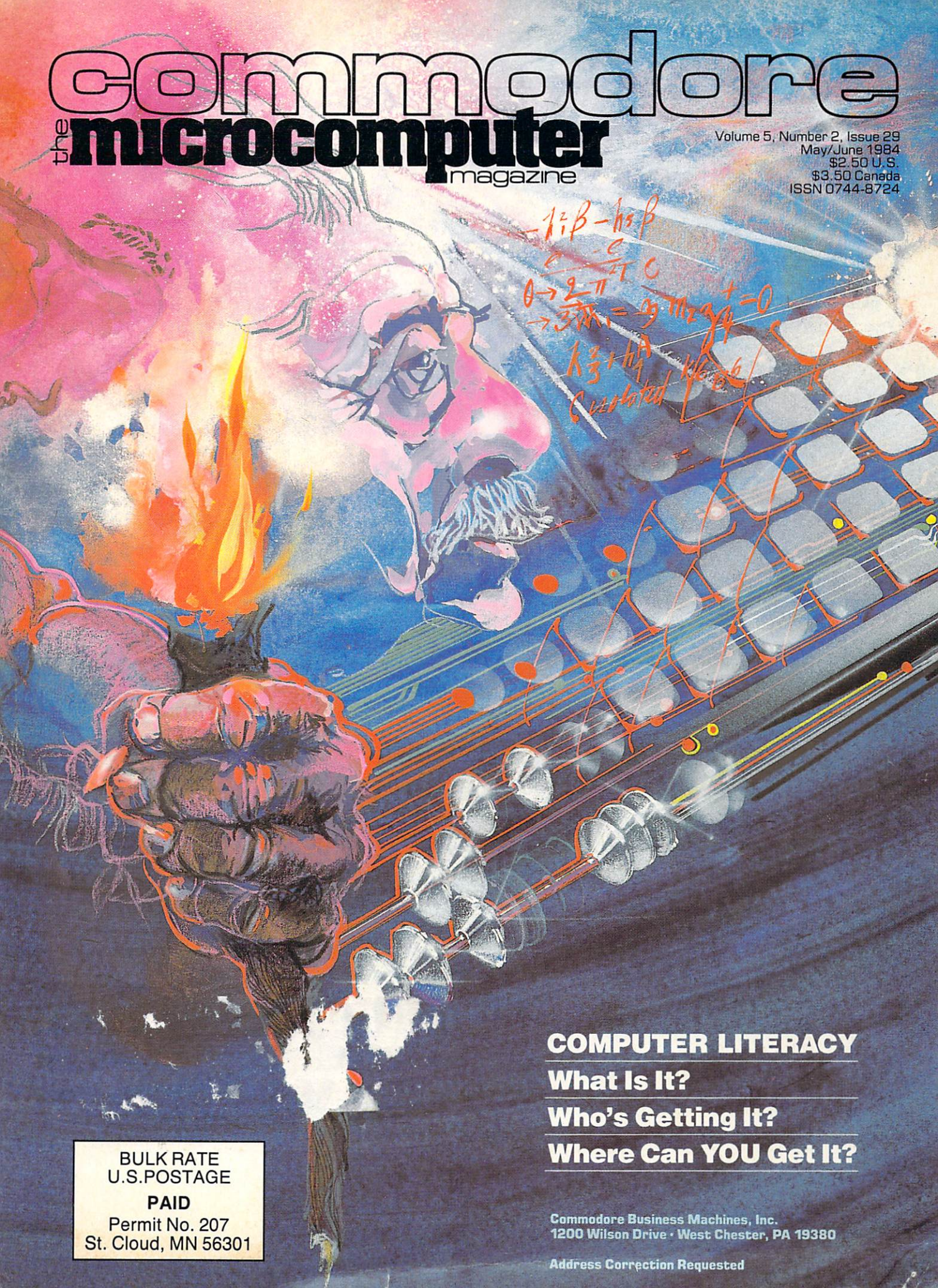
commodore
the microcomputer magazine

Volume 5, Number 2, Issue 29
May/June 1984
$2.50 U.S.
$3.50 Canada
ISSN 0744-8724



COMPUTER LITERACY

What Is It?

Who's Getting It?

Where Can YOU Get It?

BULK RATE
U.S.POSTAGE
PAID
Permit No. 207
St. Cloud, MN 56301

Commodore Business Machines, Inc.
1200 Wilson Drive · West Chester, PA 19380

Address Correction Requested

commodore 64

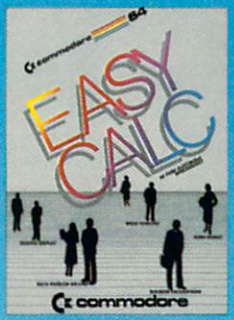
INDISPENSABLE SOFTWARE

For Your Most Important Computing Needs

Commodore is your best value in practical software—just take a look at the programs shown here—we've got everything from wordprocessing to business accounting, from electronic spreadsheets to computer graphics. Use the Software Selection Guide to find the programs which best meet your needs, then see your Commodore dealer!

EasyScript 64
Displays 764 lines × 240 characters. Prints to 130 columns. Works with EasySpell 64.

EasySpell 64
20,000 word Master Dictionary and automatic spelling checker. Works with EasyScript 64.



EasyCalc 64
Multiple electronic spreadsheet with color bar graph feature. 63 columns × 254 rows.

The Manager
Sophisticated database system with 4 built-in applications, or design your own. Text, formulas, graphics.

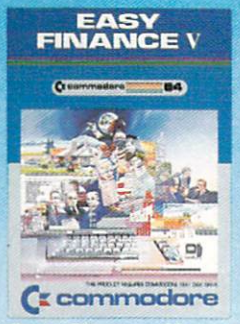
SuperExpander 64
21 special commands. Combine text with high resolution graphics. Music and game sounds.

Easy Finance I—Loan Analysis
12 loan functions. Bar graph forecasting as well as calculation.

Easy Finance II—Basic Investment Analysis
16 stock investment functions. Investment bar graph.

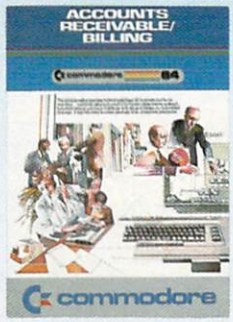
Easy Finance III—Advanced Investment Analysis
16 capital investment functions. Bar graphs.

Easy Finance IV—Business Management
21 business management features. Bar graphs.



Easy Finance V—Statistics and Forecasting
Assess present/future sales trends with 9 statistics and forecasting functions.

Accounts Payable/ Checkwriting
11 functions. Automatic billing. 50 vendors/disk.



Accounts Receivable/Billing
11 billing functions. Printed statements.

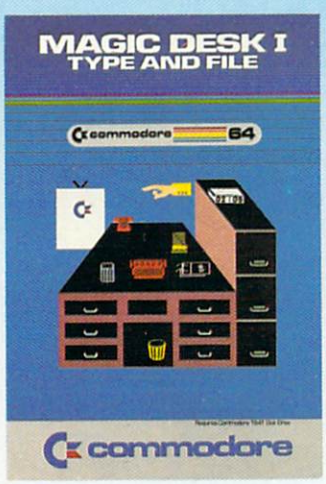
General Ledger
8 general ledger options. Custom income statement, trial balances, reports.

Inventory Management
1000 inventory items. Full reports.

Payroll
24 different payroll functions. Integrated with G/L system.

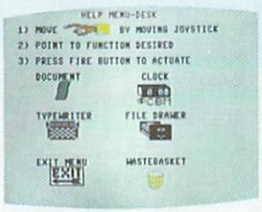
SOFTWARE SELECTION GUIDE

APPLICATION	SOFTWARE
Budget/Calculation	EASYCALC 64
Business Accounting	ACCOUNTS PAYABLE/CHECKWRITING, ACCOUNTS RECEIVABLE/BILLING, GENERAL LEDGER, INVENTORY MANAGEMENT, PAYROLL
Business Management	EASYFINANCE IV—BUSINESS MANAGEMENT
Children's Programming	ZORTEK & THE MICROCHIPS
Cooking/Recipes	MICRO COOKBOOK
Data Base Management	THE MANAGER
Electronic Spreadsheet	EASYCALC 64
Filing/Recordkeeping	MAGIC DESK, THE MANAGER, INVENTORY MANAGEMENT
Financial Investments	EASYFINANCE II—BASIC INVESTMENT ANALYSIS, EASYFINANCE III—ADVANCED INVESTMENT ANALYSIS, FINANCIAL ADVISOR
Graphics/Sound	SUPEREXPANDER 64
Learn Programming	INTRODUCTION TO BASIC—PART 1
Loans/Mortgages	EASYFINANCE I—LOAN ANALYSIS, FINANCIAL ADVISOR
Mailing List	EASYMAIL 64
Music	MUSIC COMPOSER, MUSIC MACHINE
Programming Aids	SUPEREXPANDER 64, SCREEN EDITOR, ASSEMBLER 64
Reference Books	PROGRAMMERS REFERENCE GUIDE, SOFTWARE ENCYCLOPEDIA
Spelling Dictionary	EASYSPELL 64 (for use with EASYSCRIPT 64)
Statistics/Forecasting	EASYFINANCE V—STATISTICS & FORECASTING, EASYFINANCE IV—BUSINESS MANAGEMENT
Teacher's Aids	EASYLESSON/EASYQUIZ, LOGO, PILOT
Telecommunications	VICMODEM, AUTOMODEM, TERM 20/64, RS232 INTERFACE
Wordprocessing	EASYSCRIPT 64, MAGIC DESK, WORD MACHINE/NAME MACHINE



MAGIC DESK I-TYPE & FILE

Only Commodore brings you the magic of MAGIC DESK... the next generation of "user-friendly" software! Imagine using your computer to type, file and edit personal letters and papers—without learning any special commands! All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go. MAGIC DESK is the "ultimate" in friendly software!



Special "Help" Menus

Not only is MAGIC DESK easy to use... it's hard to make a mistake! Just press the COMMODORE key and one of several "help menus" appears to tell you exactly what to do next.

commodore COMPUTERS

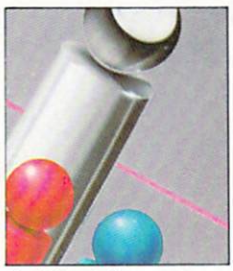
First In Quality Software

commodore
the microcomputer magazine

features

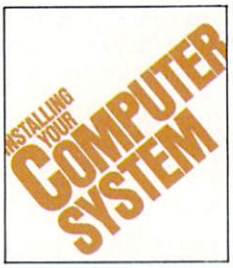
Volume 5, Number 2, Issue 29 May/June 1984

Computer Literacy



Get Literate!

Different Things



Installing Your Computer

34 The Wonderful, Slightly Screwy World of Computer Literacy by DX Fenten
A nationally syndicated computer columnist attempts to answer the question, "What is computer literacy?"

38 Get Literate! by Diane LeBold
If you want to know more about computing, you can read books, watch television, join a user group... and even go to camp.

44 Computer Literacy: Different Things to Different People
by E. Anne Winchester and Darlene Atteberry
Two educators who teach computing to all kinds of students explain who is seeking to become computer literate and what they want to know.

48 Installing Your Computer System by Joe Rotello
Create the best possible operating environment for your computer system.

52 Choosing a System: An Important Part of Computer Literacy by Jim Gracely
Our technical editor tells you how to put together a computer system that meets your particular needs.

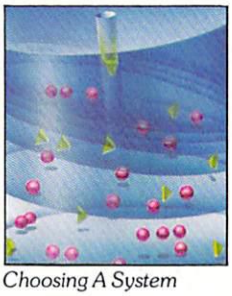
56 Books: Still a Main Road to Computer Literacy by Jim Strasma
Yes, you can even read your way to literacy. The author of several books about Commodore products makes some recommendations.

departments

6 Editor's Notes

8 Industry News
Commodore Comments on Product Plans
New Commodore-Related Products from Independent Manufacturers

commodore
the microcomputer magazine


Choosing A System

Books: A Main Road

18 Law
Computer Copyright Law: *Apple vs Franklin* Leaves Many Questions Unanswered by Herbert Swartz

21 Education
A Look at *Kinder Koncepts* by April Koppenhaver
Using the VIC With Exceptional Individuals by Arthur J. Spring, Jr.
No-Nonsense News About Networking by Pat Walkington, Gail Austin and Roger Lindholm
Educators Learn Computer Literacy at Vermont Resort

29 Computer Languages
Captain COMAL™ Meets the Disk Directory by Len Lindsay

69 Programmer's Tips
Translating Machine Language Routines into BASIC Loaders by M. W. Caprio
Random Thoughts, Part 5: The Poisson Distribution by Mark Zimmermann
250-Digit Arithmetic by William Miller

87 Technical
Curve Fitting Simplified by Bryan P. Herve

92 User Departments
Commodore 64
Smooth Scrolling Left-to-Right by Brian Schott
Standard Screen Plotting by Rolf L. Miller
Indexer by W. J. Crowley
PET/CBM
Multi-File ASCII-to-WordPro Conversion by Elizabeth Deal
Delayed Shell Sort by Joe Rotello
SuperPET
SuperPET Potpourri by Dick Barnes
B Series
The B Series Machine Language Monitor by Howard Rotenberg
Programming B128 Function Keys by Al Fragola

116 User Groups
User Group Listing
Find the Commodore user group nearest you.
User Bulletin Board

124 Product Review
***The Manager* for the Commodore 64**
reviewed by Thomas Ziegler

128 That Does Not Compute
When we make a mistake, this is where we fix it.

128 Advertisers Index

staff

Director—Publishing
Neil Harris
Editor
Diane LeBold
Technical Editor
Jim Gracely
Staff Writers
April Koppenhaver
Howard Rotenberg
Pat Walkington
Thomas Ziegler
Contributing Writers
Darlene Atteberry
Gail Austin
Dick Barnes
M. W. Caprio
W.J. Crowley
Elizabeth Deal
DX Fenten
Al Fragola
Bryan P. Herve
Roger Lindholm
Len Lindsay
Rolf L. Miller
William Miller
Joe Rotello
Brian Schott
Arthur J. Spring, Jr.
Jim Strasma
Herbert Swartz
E. Anne Winchester
Mark Zimmermann
Technical Staff
Tony Caramanico
Barbara Karpinski
Circulation Manager
John O'Brien
Circulation Assistant
Kathy Reigel
Advertising Sales Manager
Pamela S. Fedor
Advertising Coordinator
Sharon Steinhofer
Graphic Design
Neumann Greenberg Schlenker
King of Prussia, Pennsylvania
Cover
Robert Neumann
Typography
Associates International, Inc.
Wilmington, Delaware
Printing
Volkmuth Printers
St. Cloud, Minnesota

ISBN 0-88731-008-7

VIC 20™, Commodore 64™ and SuperPET™ are trademarks of Commodore Electronics Ltd. PET® is a registered trademark of Commodore Business Machines, Inc. CBM® is a registered trademark of Commodore Electronics Ltd.

Commodore: The Microcomputer Magazine is published by the Computer Systems Division, Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, Pennsylvania 19380. Copyright 1984© by Commodore Electronics Ltd. All rights reserved. No material may be reprinted without permission. Volume 5, Number 2, Issue 29: May/June 1984.

Subscription Orders: Phone 800-345-8112 (In Pennsylvania 800-662-2444).

Watch for These Upcoming Issues

Power/Play, Issue 9 (June/July): Our little sister has changed her style and gone bi-monthly, and does she look GOOD! Check out our Strategy Game Special Issue, on its way to you in mid-May.

Commodore Microcomputers, Issue 30 (July/August): Our next issue features telecommunications—who's doing it, what they're doing, where, how and why—and how you can get in on this next wave of computing. A crucial issue for those who want to stay on the leading edge. Don't miss it!

Key to Entering Program Listings

```
"[F1,F2,F3,F4,F5,F6,F7,F8]":F1,F2,F3,F4,
  F5,F6, F7 AND F8
"[POUND]":ENGLISH POUND
"[PI]"PI SYMBOL
"^":UP ARROW
"[HOME]":UNSHIFTED CLR/HOME
"[CLEAR]":SHIFTED CLR/HOME
"[RVS]":REVERSE ON
"[RVOFF]":REVERSE OFF
"[BLACK,WHITE,RED,CYAN,MAGENTA,GREEN,BLUE,
  YELLOW]" THE 8 CTRL KEY COLORS
"[ORANGE,BROWN,L. RED,GRAY 1,GRAY 2,L.
  GREEN,L. BLUE, GRAY 3]":THE 8
  COMMODORE KEY COLORS (ONLY ON THE 64)
GRAPHIC SYMBOLS WILL BE REPRESENTED AS
  EITHER THE LETTERS SHFT (SHIFT KEY) AND
  A KEY:"[SHFT Q,SHFT K,SHFT V,SHFT T,
  SHFT L]"
  OR THE LETTERS CMDR (COMMODORE KEY) AND
  A KEY:"[CMDR Q,CMDR H,CMDR S,CMDR N,
  CMDR O]"
IF A SYMBOL IS REPEATED, THE NUMBER OF
  REPITITIONS WILL BE DIRECTLY AFTER THE
  KEY AND BEFORE THE COMMA:"[SPACE3,
  SHFT S4,CMDR M2]"
```

Be confident because your PowerType Daisywheel correspondence becomes you.

When you can't personally be there, only the clearest typed correspondence should be your substitute. With a PowerType Daisywheel printer your documents look highly professional. And so do you!

PowerType. It's "typewriter friendly." Using a simple drop-in ribbon cassette, it bi-directionally types executive quality correspondence at 18 cps with a print wheel that holds 96 flawless characters.

Designed for personal or business applications, PowerType's carriage accepts paper that ranges from letter to legal size, from fanfold to roll to cut sheet. You can set right and left margins, vertical and horizontal tabs.

Plus, of course, PowerType has both serial and parallel interfaces to enable it to connect to just about any personal or business computer.

So the next time you're going face to face through the mail, rely on PowerType. It will help you make a professional impression. And that's always very becoming.

star micronics·inc™

THE POWER BEHIND THE PRINTED WORD.

Computer Peripherals Division

P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261 (214) 456-0052

If You're Reading This, Does That Mean You're Literate?

Although this may come as no news to our readers, it occurred to me as we were putting this issue together that computer literacy, our feature topic, is really what our user magazines are *always* about. Every single thing that goes into both *Power/Play* and *Commodore* magazines is designed to help our readers learn more about their computers—to make them more literate. So the fact that we happened to call this the "Computer Literacy" issue is more or less beside the point. *Every* issue is a computer literacy issue.

Nevertheless, it was fun to put together a concentrated overview of the who-what-where-why-and-how's of computer literacy, 1984. It made us take a closer look than usual at what our readers really need to know and the ways we can get that information to them. When you're around computers all the time, you see, it's easy to start assuming that everyone in the world understands at least certain fundamental concepts of computing. But only a short time ago I was talking blithely about different types of software to a friend who stopped me to ask, "Exactly what *is* software, anyway?" Poof! So much for "everyone in the world".

Those of you who have been reading the *Wall Street Journal* and *New York Times* may be wondering what's going on with Commodore's new 264/364 computers. Commodore's chairman of the board, Irving Gould, addresses your questions in our new "Industry News" section, beginning on page 8.

Speaking of the "Industry News" section brings us to an important point. We used to call that department "Commodore News", but now we're broadening our horizons to bring you more comprehensive coverage of what's going on in the industry as a whole. You'll also be seeing other changes in our format and graphic design that we think you'll like. We'll be adding a few new items, including in-depth reviews of products from independent manufacturers, and reorganizing our pages to make articles easier to follow and better looking than ever before.

Our sister publication, *Power/Play,* which was formerly a quarterly publication, is going bimonthly as of the June/July issue and will focus more specifically than ever before on games and recreation. *Commodore* magazine will continue to cover the more "serious" side of computing—education, business, the arts and more advanced programming—and will continue to get into your hands about the middle of the even-numbered months, although (for the sake of newsstand sales) the cover dates will be the odd-numbered months. So, for instance, you got this issue in mid-April as usual, but we're calling it May/June, right? Don't panic. You didn't miss any issues.

In this issue, let me draw your attention to Herbert Swartz's article on computer copyright law. The recent Betamax decision from the Supreme Court, which says, in short, that it's okay to tape TV shows for your own use, affects not just the VCR industry but the computer industry as well, since computers and their programs fall into the same weird never-neverland as video tapes when it comes to copyright law. If you can copy TV shows, can you then legally copy software? What about operating systems? In the *Apple vs Franklin* case Herbert discusses here, the U.S. Court of Appeals said operating systems are copyrightable, but will the Supreme Court decision now cast doubt on that? The lumbering old legal system just has not been able to keep up with high flying technology.

Next issue we'll be focusing on telecommunications. The opinion around here is that more and more people will be using computers to access information over the telephone lines, and that, in fact, telecommunications is the wave of the future—in which case Commodore users, with their inexpensive modems, are going to be riding the crest, shooting the curl, hanging ten... and whatever other "wave" images may come to mind. For those of you who aren't quite sure what "telecommunications" is—we'll help you get educated in the July/August issue. C

—Diane LeBold
Editor

Commodore Indispensable Software... For When You Really Mean Business

Computer systems don't have to be expensive to be effective—even in business. Commodore has been proving that for years.

Now your Commodore 64 combined with a full-range of application software means business when you do.

Whether you're trying to run a small business or manage your home finances, the Commodore accounting series software makes it affordable and easy.

Included in the series are the following software packages:

The **GENERAL LEDGER** program brings you into the computer era with an economical system for maintaining and monitoring profit and loss records, balance sheets and financial reports.

Monitoring payables and receivables is always a problem for any small business. Commodore's **ACCOUNTS PAYABLE/CHECKWRITING** and **ACCOUNTS RECEIVABLE/BILLING** computerize your invoicing, customer sales and credit information. Handling up to 50 vendors or 75 customers, and maximum monthly totals of $99,999,999.99, these two programs are truly indispensable.

The series also includes a comprehensive **INVENTORY CONTROL** program. With over 1000 items per disk, it's perfect for retail stores, mail order businesses, libraries, bookstores, warehouses—virtually anywhere large numbers of items need to be tracked, priced, billed, monitored and recorded.

When it comes to paying your employees, Commodore also solves that problem. Our **PAYROLL** package allows for up to 50 people and automatically computes tax calculations.

For further details on these packages and information on the complete line of application software, contact your local Commodore dealer.

The complete Commodore 64 Accounting series software—**Sophisticated business packages at an Unsophisticated price!!**

commodore COMPUTERS

First In Quality Software

Commodore Comments On Product Plans

Mr. Irving Gould, Chairman of the Board of Commodore International Limited, issued the following statement concerning Commodore's current and future product plans.

"When the Commodore 64 was first introduced into the mass market one year ago, we anticipated it would be very well received. Its reception and continuing strength to date has exceeded our most optimistic projections.

"When we introduced the new Commodore 264, we recognized that our Commodore 64 had and was continuing to receive exceptional consumer, educational and small business acceptance in the marketplace and that it was, by far, the best selling microcomputer in the world.

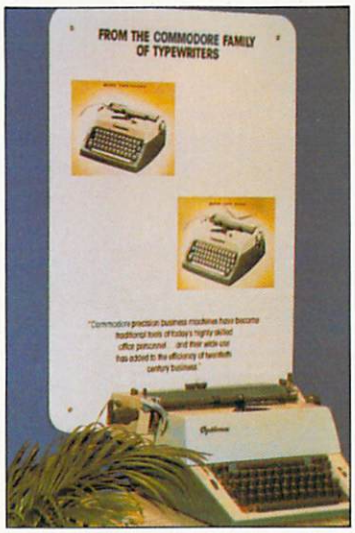
"At the present time, the demand for the Commodore 64 continues at a more rapid rate than we had anticipated, a rate of demand that exceeded our production capabilities even during the post-Christmas quarter. As a result, we are continuing to increase our Commodore 64 production capacity.

"As for our plans related to the Commodore 264, this new microcomputer is still planned to be introduced this year at a time when our capacity permits both the continuing increase in Commodore 64 production as well as large scale production of the Commodore 264.

"In short, we had anticipated excellent demand for the Commodore 64 and what we are now experiencing is extraordinary demand not only for this microcomputer, but also for the peripheral equipment and software that accompanies it."

Original Commodore Products Shown at 25th Birthday

The one and only original PET, first shown at the January, 1977, Consumer Electronics Show in Chicago.



Commodore began as a typewriter business.

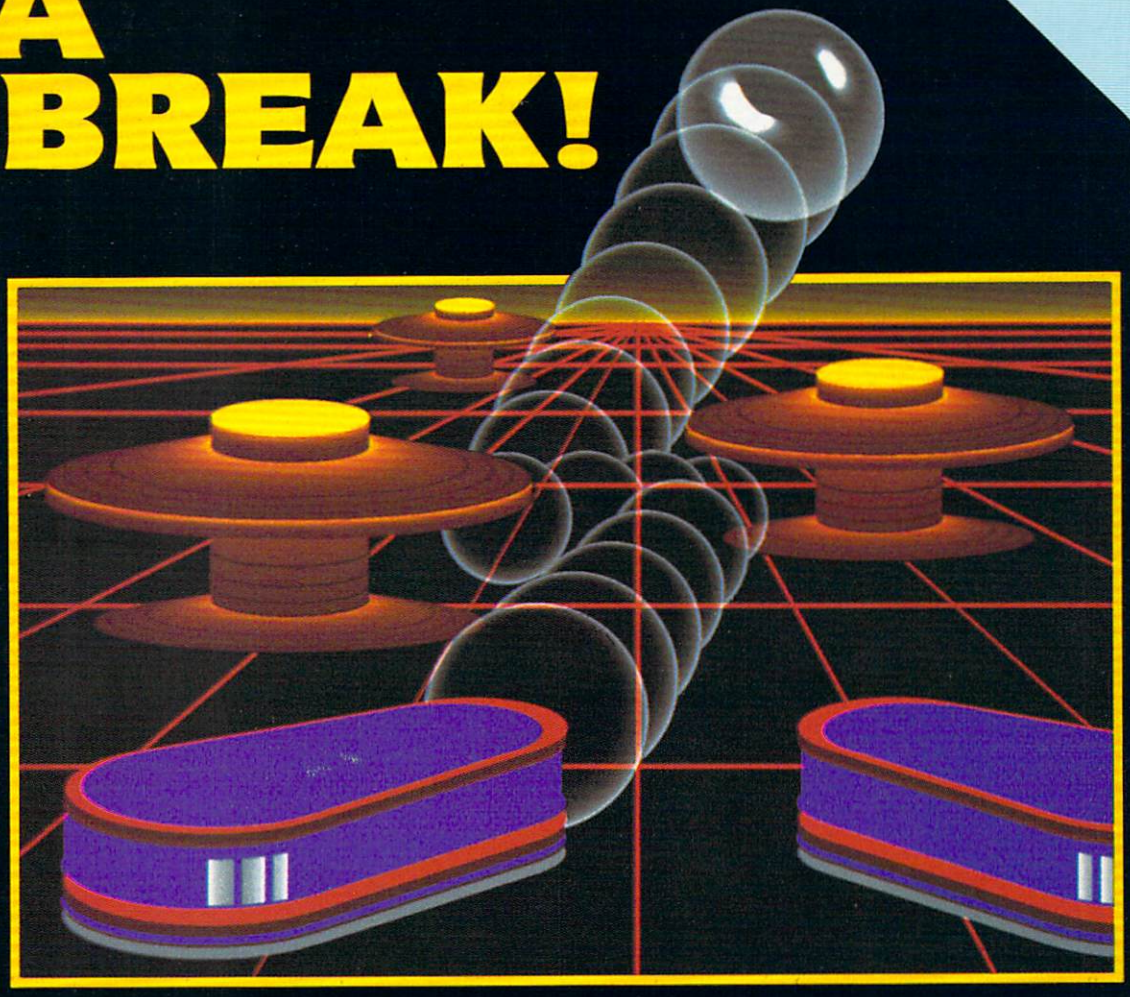
At Commodore's 25th birthday celebration in Toronto last December, the company had a sampling of some of its original product lines on hand. Neil Harris, *Commodore* magazine's director of publishing, snapped these photos of a few of the old products, including *the original* PET computer, whose case, believe it or not, was made out of wood.

In addition to the typewriters and calculators shown in the photos here, Commodore also distributed radios and manufactured digital watches.

Electronic calculators were one of Commodore's main products for many years.

TAKE A BREAK!

For Commodore 64



WITH NIGHT MISSION PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of **Night Mission PINBALL**, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put **Night Mission PINBALL** in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with **Night Mission PINBALL** from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.

See your dealer . . .

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820 USA
(217) 359-8482 Telex: 206995

industry news

Computers for the Handicapped

A national conference to examine the impact of microcomputer technology for the handicapped will be held in Minneapolis September 13-16, 1984.

Titled "Computer Technology For The Handicapped," the conference will be held at the Radisson South Hotel. Sponsors are *Closing the Gap* (an international newspaper covering microcomputer applications for the handicapped) and TAM (Technology and Media, forming as a division of the Council For Exceptional Children) in cooperation with Dr. Gilbert Schiffman, Department of Education, Johns Hopkins University and Dr. William Buchanan, Applied Physics Laboratory, Johns Hopkins University.

This conference and its program will bring together special education, rehabilitation and medical professionals from around the world to share advances made by technology to aid the handicapped. In addition, the conference is designed to provide information to parents of handicapped children and disabled individuals, demonstrating the opportunities, advances and support technology can bring to this population.

Over 80 presentations and three three-hour workshops are available to all participants. In addition, there are six pre-conference workshops featuring beginner to advanced training in the use of microcomputers for the handicapped.

A nationally-based commercial exhibit floor featuring the latest technology has to offer will also be a part of the conference and will be available to the general public for a $3.00 admission fee.

Total cost of the conference is $150 if registration is received before September 1st, or $175 for registration after that date. TAM and Group discounts are also available.

Additional information may be obtained from *Closing the Gap*, P.O. Box 68, Henderson, MN (612) 665-6573 (Metro Mpls./St. Paul, 341-8299).

ADDON Increases Versatility of Commodore 64

Quality Computer recently announced the release of ADDON, a ten-key numeric pad for the Commodore 64. ADDON's design allows for quick tabulation of figures and arithmetic functions. It is also simple to attach to existing hardware.

In addition to the above keys, ADDON features omni-directional keys for greater versatility.

ADDON's retail price of $79.95 is considerably below most comparable ten-key pads. Dave Stewart, creator of ADDON, explains the company's reasoning behind this pricing. "We feel it is important for users to own ADDON because it cuts down the time needed for numerical input. Personally, I can't imagine entering numbers without it. As a real computer buff, I can really appreciate its benefits."

ADDON can be purchased through Quality Computer in Ventura, California, or local Commodore dealers.

PETSpeed User Guide Available

CompuSystems Management announces the PETSpeed User Guide, an in-depth instruction and informational manual directed to both first time and advanced users of the PETSpeed line of Commodore-compatible BASIC compiler programs. The User Guide covers such subjects as an in-depth introduction to compilers and PETSpeed in particular, terminology, PETSpeed major operating sections, how PETSpeed treats your BASIC source code, error codes, basic and advanced PETSpeed programming via compiler machine language interaction and a comprehensive detailed chapter on useable sample source code programs in addition to many other subjects.

The User Guide is applicable to all Commodore 64, 4000 and 8000 series of Commodore computers, including exclusive coverage of PETSpeed for the new generation Commodores soon to be released. The User Guide is in an easy-to-read and use 8.5" by 11" comb-bound format with charts and listings facing text pages for easy reference. Availability is first quarter 1984. Pre-paid reserve orders are now being accepted at a special pre-publication rate. Please contact: Joe Rotello, CompuSystems Management, 4734 East 26th Street, Tucson, AZ 85711, (602) 790-6333.

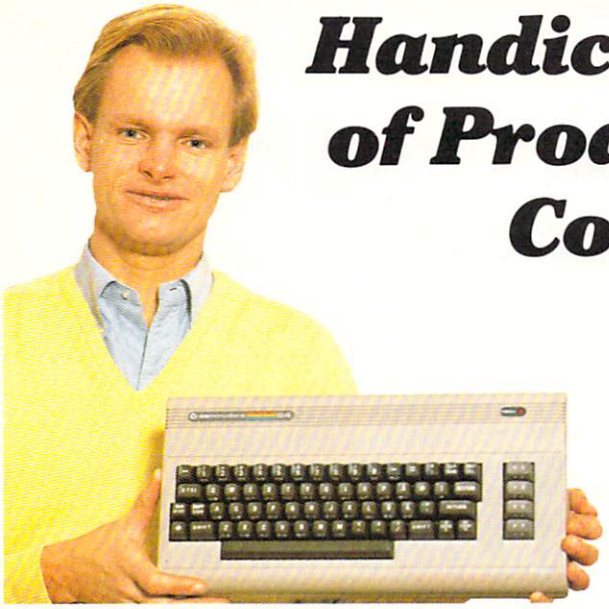
Mad River Video Surges Ahead

The leading edge in telecommunications is being established by Mad River Video, a small company in Vermont. It is the first company in the nation to use computers to take advantage of a recently deregulated FCC communications frequency. The frequency, called SCA (subsidiary communications authorization), allows FM radio stations to broadcast television channels of videotext and graphics along with their regular signal.

Starting at the end of 1983, Mad River Video and radio station WNCS, 96.7 FM in Montpelier, Vermont, started a subscription television service. This service, called The GRAIL (Greater Resort Area Information Link) is a one-way videotext and graphic channel for resorts throughout central Vermont. Commodore 64's are used exclusively throughout the system.

This channel will provide complete recreational information to subscribers year round. The information includes everything from skiing conditions to locations of covered bridges. The regular audio broadcasts of WNCS (a contemporary adult music station) will also be received on the televisions of users subscribing to the GRAIL channel.

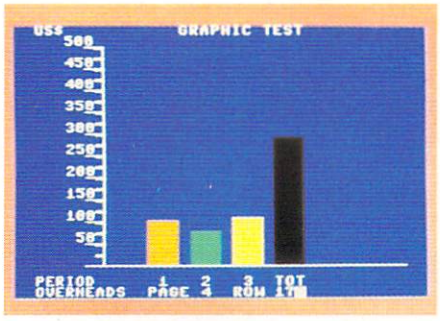
Over 500 resorts have already hooked into this service in Stowe and Mad River Valley, with more joining constantly. There are 40 pages or screens of information, which are displayed for 25 seconds each.



Handic – A Complete Line of Products to Keep Your Commodore 64 Busy – Everyday!

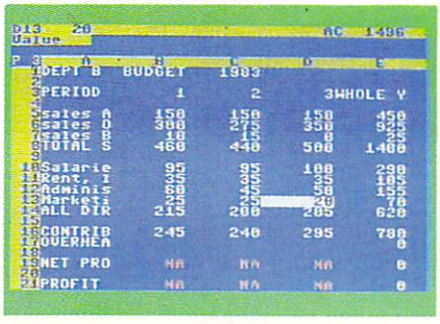
Handle your home budget, stock portfolio, loans and mortgages with Calc Result

Calc Result Easy is a simple-to-use spreadsheet program for the Commodore 64. It includes 254 lines × 64 columns, built-in graphics, and flexible printout formats. Plug-in cartridge... just plug it in and its ready. Perfect for cash flow analysis, personal net worth, IRA analysis, travel expenses, credit card expenditures, gas and electricity bills, etc.



Calc Result Easy $49.95

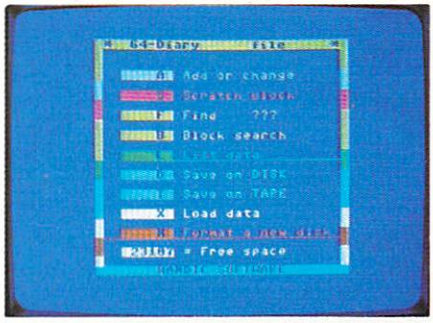
Calc Result Advanced gives you 32 pages of interrelated information. The three-dimensional feature allows you to consolidate calculations in summary format. Calc Result Advanced comes on plug-in cartridge and disk. Disk drive required.



Calc Result Advanced $99.95

A complete database for the home

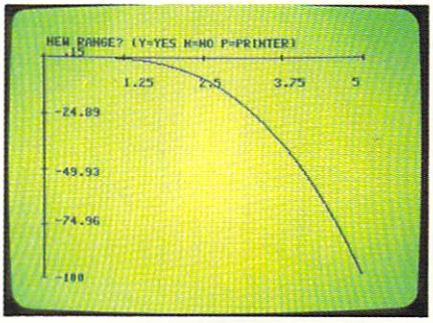
Addresses, telephone numbers, appointments, birthdays, or records–whatever you want to remember–put it on DIARY, an electronic notebook for home use. DIARY comes on a plug-in cartridge. It's easy to use and easy to learn, giving you the flexibility to design a personal calendar or address book.



Diary $29.95

Turn statistical information into graphic format

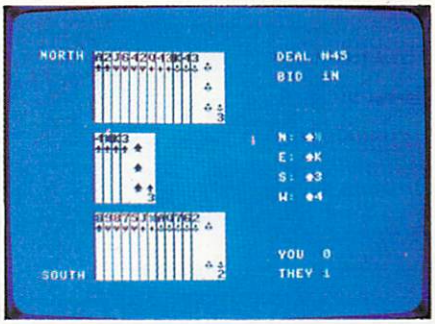
GRAF 64 converts mathematical functions into graphical analysis on the Commodore 64. An ideal program for studying math. Define a function, set the limits of an axis, plot a graph and display the extreme points, intersection values, etc.



Graf 64 $29.95

Develop your bridge skills

Whether you're an experienced bridge player or a beginner, polish your skills or learn the game with BRIDGE 64. Play North-South, then switch to East-West in the same deal, the return to that deal again and test your skill with a different strategy.



Bridge $39.95

Handic–for the broadest range of Commodore products

As the largest independent developer of Commodore software and accessories, Handic's broad range of business, education and recreation products are designed exclusively for the Commodore user who demands quality and reliability.

For more information and a catalogue of our products, see your nearest Commodore dealer, or call us direct.

handic software

Handic Software, Inc.
Fellowship Business Center
520 Fellowship Road, B 206
Mount Laurel, NJ 08054
Phone (609)663-0660

Commodore 64 is a registered trademark of Commodore Electronics, Ltd.

industry news

Contest for Computer Songsters

If you dabble in computer music, you may be interested in the First Annual Computer Songwriting Contest. Sponsored by Entech Software, the contest is designed to show off their new music package called STUDIO 64 for the Commodore 64.

The songs submitted must be original compositions written with the STUDIO 64 package (what did you expect?). There are limits of five minutes per song and one song per disk, but you can enter as often as you like. Songs will be judged by a panel of music professionals who will be scoring songs in the areas of originality, creativity and use of sound.

The contest is open to residents of the U.S.A. and Canada and offers some impressive prizes. The first place prize is $1000 cash plus studio time, including studio musicians (apparently to record your winning song). Second prize is $500 with the same studio time and third prize is the studio time alone.

The deadline for submissions is November 1, 1984, with a winner announced on December 15, 1984. There will also be an awards ceremony at the 1985 Winter CES show.

Anyone wishing official rules and an entry form are asked to write to: Computer Songwriting Contest, P.O. Box 881, Sun Valley, CA 91353.

Modern Alternative to Pencil and Paper

Inkwell Systems, the creators of Penware™ products, has recently introduced FLEXIDRAW™, a versatile graphics software program coupled with a high-performance light pen for the Commodore 64 computer. The FLEXIDRAW package enables 64 users to perform pencil-and-paper routines utilizing the speed and full graphic capabilities of their computer. As a result, the 64 user can produce anything from simple free-hand sketches to complex CAD-type drawings.

FLEXIDRAW, the first in a series of Penware products, features a menu of automatic graphic selections such as instant line (point-to-point), box, circle and rubber band choices, zoom for detailed work, two separate work areas and Put-Get commands for manipulating images on the screen or transferring them between the two areas, and shading and pattern fills for added graphic variations.

All graphics generated with FLEXIDRAW can be stored on disk for later use or printed to hard copy. FLEXIDRAW can be easily interfaced with Commodore's 1525, Epson's MX70, the Gemini 10/10X, C. Itoh's Prowriter and the Okidata 84. The FLEXIDRAW program has a full 90-day warranty.

The accompanying light pen is durably constructed and has a two-year warranty. It performs well on standard TV's as well as color, black/white and most green-screen monitors. Screen response time is pegged at 175 nanoseconds with a two-line resolution capability.

Inquiries can be directed to Inkwell Systems by calling (619) 268-8792 or by writing them at 7760 Vickers Street, P.O. Box 85152 MB290, San Diego, CA 92138.

New Light Pen from Madison

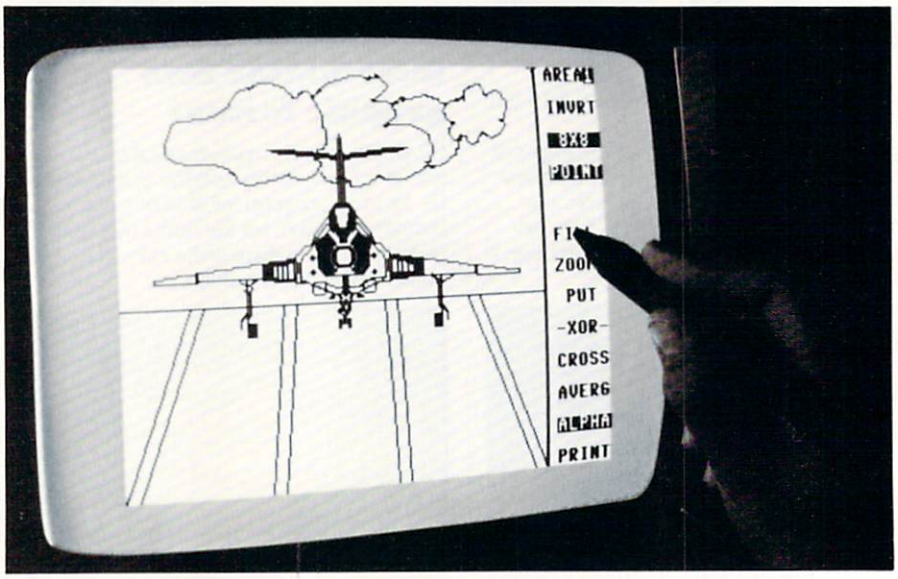
Madison Computer introduces the remarkable MCPEN™—a high-resolution light pen that allows you to answer questions, create drawings, choose options and play games by simply pointing it at the screen.

What sets MCPEN apart from other light pens is its high resolution, which permits both vertical and horizontal precision. MCPEN has two other unique features that make it especially easy to use. First, the sensitivity control on the stand allows you to fine tune MCPEN's responsiveness to suit your individual preference. As a result, MCPEN is able to adapt to a variety of monitors, brightness levels and room lighting levels. Second, while using MCPEN, you retain full functional use of your keyboard.

MCPEN rests in an attractive stand, and is attached by a long, lightweight cord to permit freedom of movement. MCPEN was designed for use with the VIC 20, Commodore 64, and Atari computers and interfaces with the computer via the joystick port.

MCPEN retails for only $49.95, complete with stand, easy-to-use manual and software. The manual provides a detailed explanation of how the device operates, as well as all of the necessary information for writing your own exciting programs for MCPEN.

MCPEN is available through: Madison Computer, 1825 Monroe Street, Madison, WI 53711, (608) 255-5552.



Inkwell's FLEXIDRAW for the 64.

Flight Simulator II

For the Commodore 64™

Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

industry news

NEW COURSEWARE AUTHORING SYSTEM

Tough Technologies, Inc. announced the release of an easy to use, powerful educational courseware authoring system. CLAS, which stands for Computerized Lesson Authoring System, is a cleverly designed tool that helps educators and parents cope with mounting "computer pressures" by enabling them to create their own courseware without any computer experience. With CLAS, teachers can confidently and comfortably utilize the latest computer technology and yet retain their individual style and teaching methods.

CLAS is not just a test writer. It can be used to teach elementary school children or college graduate students anything from basic math and spelling to physics and technical writing. CLAS can be used to teach a single lesson on a single topic or an entire semester's course. It can be used as an aid in teaching or as an independent learning tool.

A CLAS lesson can take many forms, not just drill and practice simulation. It can consist of reading materials followed by problem sets made up of multiple choice, fill-in-the-blanks, and/or true-false questions. And the reading material itself can contain fill-in-the-blanks questions. Problems can be presented in random or prescribed order. Materials can be easy, difficult, humorous or rigorous.

CLAS is currently available for most microcomputers, including the Commodore 64. The system includes an Author disk, a Student disk and a comprehensive User's Guide and is priced at $89.95.

New Commodore 64 Programs from KIDware

KIDware, manufacturer of low-cost, high-quality software just for kids, has released its first offerings for the Commodore 64.

Over 20 programs are available for children one to 16 years old. Each program is non-violent, inexpensive, educational and fun. Every program highlights the powerful graphics and music of the Commodore 64 and all are extensively KID-tested.

Programs can be obtained on cassette tape or disk. Each KIDware cassette or disk has not one, but *two* complete programs. And the price for these two-program packages is just $10.95 (cassette) and $13.95 (disk).

For info, contact: KIDware, P.O. Box 1664, Idaho Falls, Idaho 83403, (208) 529-9916.

Compose Music on the 64

A new software package that transforms the Commodore 64 QWERTY keyboard into a musical keyboard that can be played in any desired scale or melodic sequence has been introduced by Waveform Corporation.

The new program, called *MusiCalc 3 Keyboard Maker,* can be used with or without the previously-announced *MusiCalc 1 Sequencer and Synthesizer. MusiCalc 3* can have the same effect on the QWERTY keyboard as a capo has on a guitar. It allows the user to play in different keys without changing position fingerings.

According to Thomas A. McCreery, Waveform's President, "*MusiCalc 3* doesn't require any prior experience or understanding of standard music notation or music theory. The *MusiCalc 3* package contains a simple diagram that shows non-musicians a simple way to convert sheet music into numbers and notes."

According to McCreery, *MusiCalc 3 Keyboard Maker* will be available through computer specialty stores, mass merchandisers and musical instrument stores. The price is $34.95.

MusiCalc 3 from Waveform.

Database Lists Over 50,000 Software Packages

"Computer software for what?! Running a funeral parlor? You must be joking."

Believe it or not, it does exist. Right along with the software to launch your own space shuttle or to help you decide on a career. How about software to manage your football team? Perhaps you would like to automate your coal mine operations.

Yes, it's true, software for nearly every conceivable application has been written. Computers have taken over many of the manual, time consuming tasks and have enabled man to devote more time to "creative" processes.

Where do you find a complete, comprehensive compilation of all of this software information? What if a programmer in England has written the software you need? How do you find him? How do you order the software? Simple. You contact MENU™, the International Software Database (ISD™).

The International Software Database is the only available on-line database of software information. The database contains information on over 50,000 software products including complete vendor and program details. This information is available in a variety of ways. For instance, a Software Report™ which is a customized search of the ISD, can be produced for a customer's needs. The ISD can be accessed on-line through Lockheed DIALOG Information Retrieval Service as File 232. Or the information can be made available for quick reference on the customer's desk in the form of a hardcopy publication titled *The Software Catalog*. *The Software Catalog* is distributed quarterly by Elsevier Science Publishing of New York.

Before buying software, even prior to a computer purchase, consultation with MENU, the International Software Database is a wise move. MENU has a broad selection of services to help every user from novice to the computer expert.

For more information call MENU toll free at (800) THE-MENU. Outside the United States or in Colorado (303) 482-5000. Orders and information requests are also accepted through the mail at MENU, The International Software Database, 1520 South College Ave., Ft. Collins, CO 80524.

Commodore 64 Gets Away from BASICs

Limbic Systems, Inc., a California-based firm specializing in computer peripherals and software for the Commodore computer line, introduces the Pascal Compiler for the Commodore 64™. The Pascal Compiler represents a major advancement in the programmability of the Commodore 64, providing a professional, business/scientific tool capable of handling complex and sophisticated applications.

"Disk-based, easy to operate, the Pascal Compiler has ramifications within the business and educational markets," explains Limbic's Vice President, Lisa L. Carley.

The Pascal Compiler was developed by Oxford Computer Systems (Software) Ltd., authors and developers of the PETSpeed for the Commodore 8000, 4000 and 64 computers—a BASIC compiler recommended by Commodore.

The following utilities are provided as an aid to programming development: a linker (allowing for combining assembler code into the program), a debugger and an editor. The documentation provides a tutorial on Pascal program development through the compilation process, as well as offering the user an easy-access reference.

The Pascal Compiler is being distributed in the United States by Limbic Systems, Inc., and is anticipated to retail for approximately $50. Inquiries should be directed to Limbic Systems, 560 San Antonio Road, Suite 202, Palo Alto, California 94306, (415) 424-0168.

"CONTROL YOUR WORLD" WITH YOUR VIC-20

With simple circuits using low cost parts and our program supplied on cassette tape, we'll show you how to use your COMMODORE VIC-20 for:

- Digital Thermometers
- Digital Clock
- Burglar Alarm - 2 Zone, Time Controlled
- Fire Alarm - 2 Zone, Time Controlled
- Dusk to Dawn Lighting with Photo Cell
- Furnace and Air Conditioning, Clock and Thermostat
- Clock Controlled Appliance Switches

Simple program variations in basic can operate lights, motors, furnaces, machines, heat pumps, radios, sound systems, test equipment, swimming pools, garden watering, and more.

Your video screen will display simultaneously: • Two Digital Temperatures • Digital Time • Two Analog Inputs • Five Input Ports Status • Eight Output Ports Status.

GET A LOW COST EDUCATION IN COMPUTER CONTROL. ORDER YOUR CASSETTE AND INSTRUCTION BOOK NOW!

$39.90 PRICE INCLUDES POSTAGE.
Terms: MASTER CARD/VISA

The Continental Press, Inc.
Elizabethtown, PA 17022
Toll free: 800-233-0759
Collect in PA: (717) 367-1836

'PUBLIC DOMAIN'™ — SOFTWARE —

Supporting all COMMODORE computers
Written by users, for users
★ GAMES ★ UTILITIES ★ EDUCATIONAL ★

VIC 20™

collection #1 – collection #2 – collection #3
collection #4 – collection #5 – collection #6
70+ programs per collection – Tape/Disk – $10.00

VIC COLLECTION #6

3D-Maze ● A or An Quiz ● Alarm Quiz ● Alpha Sort Amortize ● Banner ● Big Six ● Bio Printer ● Breakout (J) Budget-Mgr ● Bumblebee ● Buzzword ● Cascades CBuff Peek ● CBuff Save ● Colour Pinball ● Computer Fix Coverage Predict ● Craps ● Crytogram ● Deepspace Density ● Easter Dates ● Electric Cost ● F.I.C.A. Tax Fast Sort ● FFT Calc. ● Firing Tank ● Freq. Coord. Geog. Spell Game ● Guardian ● Hi-Res ● Job Jar L Pad Design ● Labrinth ● Letter Reco ● Marston City Rvd. Match the Number ● Math Dice ● Math Fact Drill Math Test ● Math Time ● Maze Maker ● Merry Christmas Millionaire ● Mosaic ● Multiply Tables ● National Parks Nicoma ● PCB Printer ● PI Network Desig ● Quartic Quiz ● R-S-P ● Rebound Hockey ● Revenge Inst. Revenge Prog. ● Rnd. Graphics ● Rolling Sticks ● Rom Aid Shopping List ● Simon ● Slide Voice ● Snoopy Hires Spelling Aid ● Stopwatch ● Sunrise/Sunset ● Sym T Atten Tennis 3 ● Too High-Too Low ● Typing Test ● Vectors Vehicle Cost ● VIC Orbit ● VIC Symphony ● Wumpus Inst. Wumpus Prog. ● Zero Formula

COMMODORE 64™

64 collection #1 – 64 collection #2 – 64 collection #3
64 collection #4 – 64 collection #5
25+ programs per collection – Tape/Disk – $10.00

COMMODORE 64 COLLECTION #5

GAMES — Alien Capture ● Auto Rally ● Dominoes ● Explore Lost Gold Adv. ● Niche ● Qwari ● superman ● Zone X
UTILITIES — Accounting Test ● Annuity ● Average Growth Building Costs ● Buy/Sell ● Change Disk Name Cross Word ● Data Search ● Elec. Use ● Finance ● Growth Irr. Cash Flow ● Load Calc. ● Real Estate cost ● Rent or Buy
EDUCATIONAL – A or An ● Arrow Facts ● Big Math ● Element Chart Eliza ● Europe ● Hang Math ● Linear Prog. ● Loops Marblestat ● Math Facts ● Number Match ● World Quiz

PET® / CBM®

5 Utility – Tapes/Disks – $10.00 each
11 Game – Tapes/Disks – $10.00 each
6 Educational – Tapes/Disks – $10.00 each

All prices include shipping and handling.
CHECK, MONEY ORDERS,
VISA and MASTERCARD accepted.

For A Free Catalog Write:

Public Domain, Inc.
5025 S. Rangeline Rd., W. Milton, OH 45383
10:00 a.m. - 5:00 p.m. EST – Mon. thru Fri.
(513) 698-5638 or (513) 339-1725

VIC 20™, CBM® and Commodore 64™ are Trademarks of Commodore Electronics Ltd.
PET® is a Registered Trademark of Commodore Business Machines, Inc.

JOURNAL FOR MICRO-USING INVESTORS

The American Association of Microcomputer Investors, Inc. (AAMI), an independent, non-profit organization targeted at investors using microcomputers, has published the first issue of The AAMI Journal.

Published bi-monthly, The AAMI Journal provides you with:

1. In-depth articles on specific topics related to investment analysis with your microcomputer, such as using a spreadsheet program to monitor your stock portfolio or using your microcomputer for technical analysis.

2. Comprehensive reviews of investment software packages. Each software package is rated from poor to excellent on four levels: performance, documentation, ease of use and error handling. Sample reports and graphics are shown.

3. In-depth reviews of on-line stock market databases (such as, the Dow Jones News/Retrieval Service or Merlin DIAL/DATA).

4. Reviews of books and other valuable publications.

5. Information on new investment software to keep you up-to-date on what is available.

6. A listing of discounts available to AAMI members on investment software packages.

A sample issue of THE AAMI JOURNAL is available for $3.00 from the American Association of Microcomputer Investors, Inc., P.O. Box 1384, Princeton, NJ 08542, (609) 921-6494.

New Solar Energy Software Packages

A complete line of microcomputer software for solar design and energy analysis applications is now available from compuSOLAR. This company is offering three separate software packages for passive solar, active solar and photovoltaic/wind applications, which are intended for use by solar professionals and energy educators. The software is based on research done at the Los Alamos Scientific Laboratory, the University of Wisconsin, and elsewhere.

Packages are available for most microcomputers, including the Commodore 64. Package prices range from $50 to $100. A free eight-page brochure, which includes information on programs inputs and sample outputs, is available from compuSOLAR, Gum Springs Road, Jasper, AR 72641.

PHONE-DIAL 64

DIALS TOUCH TONE PHONE AUTOMATICALLY

Using your Commodore 64 WITHOUT A MODEM

Access by phone any device reactive to touch tone.

Save time when dialing long multi-digit numbers.

Generate/Call a permanent file of names and numbers.

Automatically generate/dial sequential/random numbers

AN IDEAL TIME SAVER FOR SPEED DIALING:

Church/Club, Crime Watch, Random Sales/Announcements, and other interesting uses.

DISK: (1541, 4040, 8050)

ONLY — $ 24.50
Cash, Check or Money order

INPUT SYSTEMS, INC.
25101 S. W. 194 Ave., Dept P
Homestead FL 33031,
Phone (305) 245-3141

Commodore 64
is a trademark of Commodore Electronics

The Naked Computer: Trivia and Lore

It's been said that if automobile technology had advanced at the rate of computer technology over the last 30 years, a Rolls Royce would cost $2.50 and get two million miles to the gallon.

True, unless the particular computer technology one had in mind were that of the GE Fluid computer, which attempted to use water instead of electrons for switching circuits... or the Interplex round teleprinter computer that could multiply 12 × 12 and never get anything but 143... or, the RCA BIZMAC, a vacuum tube dinosaur that took so long to build it was obsolete before it was done (it was so big that its operators wore roller skates).

These bits of trivia and other arcane facts about computerdom are the subject of a new book entitled *The Naked Computer,* available now in bookstores. According to the publisher, William Morrow & Co., *The Naked Computer* is the world's most complete treasure trove of computer lore, true facts and milestones. Hence the book's subtitle: "Layperson's Almanac of Computer Lore, Wizardry, Personalities, Memorabilia, World Records, Mind Blowers and Tomfoolery."

The hardcover book, 335 pages, is broken into 20 chapters with subject matter ranging from industry personalities and computer firsts to robotics, artificial intelligence and computer art. An extensive index is included. It is available in hardcover only, for $15.95.

For computerphiles, *The Naked Computer* will soon be available in full-text version on the nationwide DELPHI videotex system (617) 491-3393.

Stop Gambling. Start Winning. Now.

It's a fact. You will beat the dealer if you play Blackjack correctly. In Las Vegas. In Atlantic City. In dozens of foreign countries throughout the world.

They haven't changed the rules. Even multiple-deck games pose no problem if you play properly. You can win just as easily in 1984 as you could in 1961 when the first Blackjack strategies were created.

This ad is your cue to join the small group of Blackjack players who are no longer gambling. Become a strategy player and win. Consistently.

The Obstacle

Despite the wild claims made by the Blackjack system charlatans, it is not possible to learn an effective strategy overnight. Learning an effective strategy takes time and discipline. If learning a strategy were easy, everyone would be making a living playing Blackjack. As it stands, less than one percent play well enough to make money.

The Solution

BLACKJACK TEACHER simulates, in precise detail, the events that transpire in actual casino play. The display screen depicts the top view of a Blackjack table. You interact with the program just as you would an actual game. Computer controlled players occupy adjacent seats. All events occur in real-time.

BLACKJACK TEACHER teaches seven different strategies of varying complexity and accuracy. This spectrum of strategies allows you to select a strategy that suits your needs.

BLACKJACK TEACHER monitors your betting and strategy decisions (hit/stand/double/split/insurance). If your decisions are incorrect within the guidelines of your strategy, the system will display error messages showing you the correct decisions.

BLACKJACK TEACHER is the result of over ten years of Blackjack research. The strategies encompassed by the system were developed using computers. The more complex strategies are among the most powerful ever devised.

Complete documentation is included which tells you everything you need to know to become an expert strategy player.

The SOTA Story

SOTA Enterprises has consistently produced nothing less than the highest quality software. When you buy software from SOTA, we do our utmost to make sure you get your money's worth.

ATTENTION VIC 20 USERS

A new version of BLACKJACK TEACHER is now available for the VIC 20. Although not as comprehensive as the original 32K program, the VIC 20 version does teach Basic Strategy – a must for the Blackjack strategy beginner!

FILL OUT AND MAIL TODAY!

Name _____

Address _____

City _____

State _____ **Zip** _____

Make Check or Money Order Payable to:

SOTA Enterprises, Inc.
833 Garfield Ave, Suite 101
South Pasadena, CA 91030

Check Box

☐ VIC 20	($19.95)
☐ COMMODORE 64	($59.95)
☐ PET (32K)	($59.95)
☐ PET 2001 (32K)	($59.95)
☐ CBM 4032	($59.95)

Media

☐ Cassette ☐ Diskette

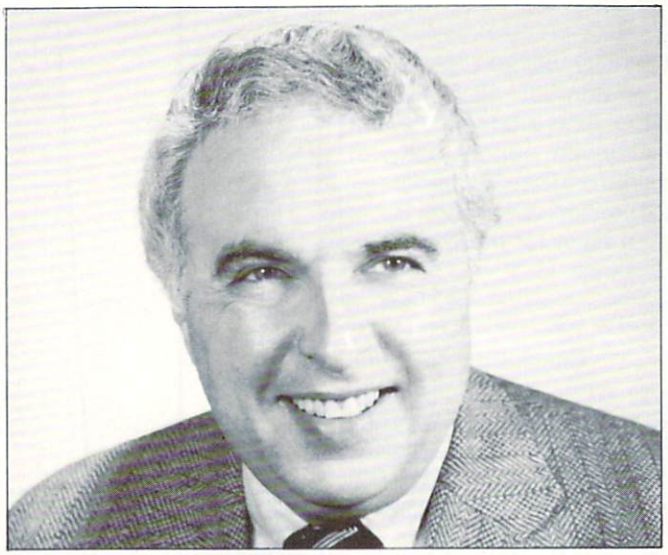
Postage Paid • California Residents add 6½% Sales Tax

Computer Copyright Law:
Apple vs Franklin Leaves Many Questions Unanswered

by Herbert Swartz

In 1982, after the Franklin computer company had begun marketing an exact clone of the popular Apple computer, Apple brought a suit against Franklin for copyright infringement. But, as Herbert explained last issue, traditional copyright law gets very murky when you try to apply it to computers and computer software. Copyright, you see, protects the "expression" but NOT the "idea" in a work. Unfortunately, when it comes to computers and their software, the place where "expression" leaves off and "idea" begins is not very clear.

As a result, Apple and Franklin battled it out for over a year. Finally, in August, 1983, the U.S. Court of Appeals for the Third Circuit (which covers Delaware, Pennsylvania and Franklin's home state, New Jersey) ruled that object code, object code in ROM and computer operational systems are, indeed, copyrightable—in the Third Circuit. But what about the other 47 states? And, given the final settlement, was it really a victory for Apple?



Copyright lawyer Herbert Swartz lives and works in New York City.

So it ends. Settled on the way to the Supreme Court. The age of the microcomputer's most celebrated law case, *Apple v. Franklin*. The parties happy. Would they ever admit otherwise?

But what of the computer industry? In retrospect, it has gained precious little, and that seems more to the point. It still has a legal basis in copyright law that is so uncertain it borders on being nonexistent.

The heralded phase of the opinion of the United States Court of Appeals for the Third Circuit—that object code, object code in ROM, and operational systems are now copyrightable—is now law in the states of the Third Circuit—Delaware, New Jersey, and Pennsylvania. This will remain the case until Congress or the Supreme Court decides otherwise. But for the other 47 states, the questions remain open. These are the anomalies in trying to run a computer industry by judicial decisions.

"The settlement marks a victory for computer manufacturers battling inexpensive copies of their own design," trumpeted the *New York Times* in the second sentence of its article on the settlement. How wrong. Franklin opted out of the litigation by paying in effect a royalty of $25 for the first 100,000 computers it manufactured. A wise business move. Further, all future disputes between the parties are to go to arbitration. No injunction can be sought by Apple to stop Franklin in its $70-million-a-year operation if at some future time Apple once again believes Franklin has infringed its copyrights.

So the industry still has miles to go before a plaintiff such as Apple can prove copyright infringement. And a host of legal questions still remain regarding what is infringement:

—Copyright protects the expression, but not the idea in a work. If idea and expression *merge*, there can be no infringement because there can be no copyright. The work, in this case, cannot be copyrighted, because to allow copyright protection would then permit a limited monopoly on *ideas*. But we still have no legal guidance as to what to do when such a merge occurs. If object code, object code in ROM and operational systems can be expressed in only one way, we have a merger of expression and idea. But the issue of copyright protection when merger takes place remains totally unresolved in computer law as it now stands.

—When does a program in ROM become *utilitarian* rather than *expression*? A utilitarian object cannot be infringed, never mind copyrighted and here, too, the case leaves us completely without help.

—Clearly a program can become so standardized that it changes from an expression to the level of a procedure. A procedure too cannot be copyrighted or infringed. But when does this come to pass? Again, no assistance from the case.

—There is, further, for copyright protection of software the issue of "*scenes a faires.*" "Stock expression" is the way courts translate the doctrine. Another copyright exception. Another area of legal darkness.

—Most important of all, what of "second-generation copyright" as the issue is called by former Copyright Office general counsel Jon Baumgarten, now a partner in the Washington/New York firm of Paskus, Gordon & Hyman?

Franklin, if indeed it had committed copyright infringement, had copied 100 percent of the Apple programs in some cases. Such copying is wrongful —as with photostating, VCR recording or copying a disk. That part is easy, assuming the Supreme Court doesn't drop a bomb with the Betamax case.*

But suppose, as is most likely to be the case with computer programs, the copying is partial—the situation of "second-generation copyright." The law allows partial copying up to the level of "substantial similarity." Actually, our scheme of copyright encourages it. The "ultimate aim," says the Supreme Court, "is to secure a fair return for an author's creative labor" so as "to stimulate artistic creativity for the general public good." Adds famed Harvard Law School professor and copyright expert Zechariah Chaffee, Jr., "A dwarf standing on the shoulders of a giant can see farther than the giant." Where then is "substantial similarity" in the computer universe?

Picture a meeting of computer executive, ace marketer, hot-shot techie and general counsel, all sitting down to plan marketing strategy. The point is to do as much as the law allows—that is the duty to shareholders—while, of course, not committing infringements. Where lies the dividing line? How much to copy of a competitor's work? How much to expand upon it?

Too far is folly, business and legal. Not enough is foolishness, not paying attention to state-of-the-art. So is 20 percent permissible? Or 40, or 60 percent? How do you even measure "substantial similarity" with computer software? Are there differences when hardware is involved? A computer is an "article of commerce," as is Franklin's. Will courts bar such articles when they include in part some copyrighted material—say, object code embedded in ROM? Probably not—because a free marketplace is the goal. Is this thus a method to circumvent the issue of copyright infringement through computer technology?

We are all still at sea. "All the major questions remain unresolved," says former Copyright Office

*__Ed. Note:__ *The Supreme Court did indeed drop "the bomb", Herbert informed us at press time, when it announced its decision in the Betamax case, saying, in short, that it is* legal *to record TV programs on VCR's. In a future issue he will discuss what that controversial decision means to the computer industry.*

counsel Baumgarten, who was retained by Franklin after the adverse decision in the Third Circuit to work on the appeal to the Supreme Court. "This settlement doesn't change that fact."

How silly then, too, *The Times* report from Arizona attorney Jack Brown, who represented Apple: "We think it was a total victory... The message has gone out that companies involved in copying will simply not be tolerated." But the point is that copyright law still blesses *some* copying. The issue is, where are the lines?

The Franklin executives are a class act. Formerly executives at the best computer companies, they acted on the advice of their lawyers, testified to the fact under oath. No slippery types here from a land across the body of water due west of the bar at the Beverly Hills Hotel.

So plenty of confusion remains for our hypothetical meeting. To say that, "The message has gone out that companies involved in copying will simply not be tolerated," is dead wrong. *Levels* of copying is what our meeting is about, because no "total victory" occurred in the Apple/Franklin case. The *next* Franklin still doesn't know how far it can go; the next Apple doesn't know when it can be sure of prevailing if it sues.

Apple and Franklin have agreed to arbitrate future differences, so a wise lesson from the case—maybe the wisest—is that litigation just won't do. Congress should note this fact especially. Unless that august body acts to bring legal order into copyright protection and stops deferring to the court system, we've got problems in this industry. Our guys at the hypothetical meeting know it. Most big computer companies know it—that's why they don't even copyright things like ROMs. They forget the law by outflanking it; they simply try to out-muscle everybody else in the marketplace. Sometimes that strategy works. C

PEEK
FULL FEATURE MAGAZINE
on
CASSETTE
FOR THE VIC 20™ and COMMODORE 64™
• GAMES • REVIEWS • BUSINESS
• EDUCATION • TUTORIALS • UTILITIES
• READY TO RUN PROGRAMS •
1 YEAR (12 issues) . . $49.95 6 MOS. (6 issues) $28.95
PLEASE SPECIFY VIC 20™ or COMMODORE 64™
8K Expander required for Vic 20™
subscribe today
PEEK MAGAZINE
P.O. BOX 35167
CLEVELAND, OHIO 44135
VIC 20™ and COMMODORE 64™ is a trademark of Commodore Electronics Ltd.
poke into peek
MAGAZINE

A Look at *Kinder Koncepts*

by April M. Koppenhaver
Commodore Software

Do you have an inquisitive youngster who is anxious to "play" on the family computer? *Kinder Koncepts,* soon to be available from Commodore, is simple and fun to use. Children spend hours at the computer and don't even realize they are learning valuable reading and math skills.

Developed by Michael Horner and Norm Eisenberg of Midwest Software, *Kinder Koncepts* is a series of five educational packages for preschoolers and primary grade school children. Each package has been child tested and approved! The packages are educational as well as entertaining. Even the package names created by Sandra Ritter, Product Co-ordinator, are fun and inviting. Starting from the easiest package the *Kinder Koncepts* series includes:

- Easy Match/Easy Count
- What's Next/Letters or Numbers
- A Letter Match/More or Less
- Letter Sequences/The Long or Short Of It
- Shapes and Patterns/Group It

Each *Kinder Koncepts* package contains eight lessons. In each package, four lessons teach basic reading readiness skills and four lessons teach basic math readiness skills.

"Each lesson helps prepare the child for the next step in learning how to read, add, or subtract," explains Mrs. Ritter enthusiastically. "In the 'Easy Match' lessons, the child practices shape and letter recognition. In the 'Shapes and Patterns' lessons, the child distinguishes the subtle differences among similar patterns. This is the last step before the child is ready to read. After all, the child must learn to distinguish the difference between hat and hot."

The lessons are grouped by particular concepts such as numbers, letters, shapes, etc. All lessons have a similar format and are made up of ten exercises. There are no time limits on the exercises and you have two chances to give a correct answer. Correct answers are rewarded with a happy face and fun tune. Incorrect answers receive a sad face. After two incorrect tries, the correct answer is shown and the lesson goes onto the next exercise. If your child answers eight or more of the ten exercises correctly on the first try, the program provides an entertaining musical and graphics show as a special reward.

Each program screen is brightly colored and can be easily understood by your child. For example, in "Shape Twins", the first lesson of "Easy Match", four shapes are shown and identified by 1, 2, 3 and 4. Two of the shapes are the same. Your child is to find the two shapes that are the same and press their number keys. In "Using Subtraction", the last program of the "Group It" menu, your child will see a set of rockets with launch pads below them. A set of the launch pads are hidden. Your child is to decide how many launch pads are hidden and press the matching number key.

The programs are designed so that few directions are needed. However, each package comes complete with a parent/teacher guide which details each of the lessons and suggests supplementary activities. Simply review the keyboard's letters and number keys with your child, load a lesson, and slowly go through it with your child. Your child should then be able to complete the lesson alone. With just a few words of advice from you, your child should be able to enjoy success with the programs from the very beginning.

Why is *Kinder Koncepts* so important for your child's learning and growth? Mrs. Ritter is quick to explain, "Most children need to be motivated. Fun is the number-one motivator. In *Kinder Koncepts,* the lessons are structured educationally and designed to be interesting. Children enjoy using *Kinder Koncepts.* Therefore, their attention span is increased as well as their retention of the material." Drawing on her many years of teaching experience, Mrs. Ritter continues, "*Kinder Koncepts* provides the child with the basic foundations for concrete thoughts which are involved with learning and understanding."

Will children feel intimidated by a computer?

ATTENTION DEALERS!

C.S.I. 24-Hour Turnaround...

On Commodore board repairs at an unbeatable price!

Why wait weeks for logic board repairs on Commodore 2000, 4000 and 8000 series computers.

Any logic board shipped to us UPS will be repaired and returned to you in 24-hours for a fixed charge of $50 a board. We also repair all disk drive boards except the 1541.

Our factory-trained technicians, fast turnaround and unbeatable prices has made C.S.I. the largest independent Commodore Service Center. We make Commodore Computers do what they do best. Work.

Special introductory discount!

Cut the headline from this ad and return it with your first shipment of boards and C.S.I. will give you $10.00 off on the first board we service for you.

Our customers speak for our service:

"I never thought we could get five-day turnaround on our board repairs, but we get it from C.S.I.—and that includes shipping time! They're a great supplement to our in-house service."

Dave Tyburski, Tycom, Inc.
Pittsfield, MA

"We've sent about 20 boards to them so far. The turnaround and quality has been absolutely beautiful. A great company to work with!"

Carl June, Computer Business Management
New Haven, CT

"C.S.I. has proved to us they're the quickest and cheapest anywhere!"

Steve Nasceif, Haneys Stereo & Computer Center
Detroit, MI

COMMODORES SERVICED IMMEDIATELY

C.S.I.: Also distributors of—**Eagle Computer • Sakata Monitors** • WICO Joysticks • MicroPro Software • The largest selection of Commodore 64 software—HES • M-File • Sirius • Script 64 • DATA-20 • SoftSync • PractiCalc • Paper Clip • Broderbund • Spinnaker • CompuServe • Timeworks • Sierra On-Line • Education Circuit • EPYX• Creative Software• Classic Covers • Delphi Oracle • Tronix • Memorex • Comm★DATA • XOR • CYMBAL • American Educational • Screenplay• ACCESS • Academy • Sega • Info-Design • Blue Sky Software • Cord Ltd.

C · S · I

33 Murray Hill Drive, Spring Valley, New York 10977
(914) 425-2177

education

"Not at all!" reassures Mrs. Ritter. "First of all, a computer system has a TV screen. Children are attracted to this and can relate to it. Also, children welcome the challenge that a computer offers. Working on a computer becomes an adventure. Besides, many parents already own a Commodore 64 and their children have watched them using it. Children love to imitate their parents... it's a natural that Commodore now offers software even for the youngest members of the family."

Another great feature of *Kinder Koncepts* is that each lesson automatically records your child's progress and shows it at the top of the TV screen or monitor. This Progress Record System lets you track your child's progress without sitting with him or her. It lets the child feel independent and comfortable, yet provides you with a method to monitor your child's learning. A set of three symbols is used to show your child's progress for each exercise: a full block shows the answer was correct on the first try; a half block that the answer was correct on the second try; and a dot that there were two incorrect tries.

If you see that your child almost always gets eight or more full blocks on a lesson, your child has learned the skills of the lesson. You can then suggest trying a more challenging lesson on the disk. If your child keeps getting all of the lessons on the disk correct with little or no errors, your child has mastered the lessons and should begin another disk in the series.

If your child gets several half blocks in a row, it may be helpful to explain the lesson to him or her. Then have him or her try the exercise again. Your child will usually get half blocks after he or she begins to understand a lesson but needs more practice to master it.

The *Kinder Koncepts* series is a sensational way to introduce your child to the Commodore 64 computer and the important educational skills necessary for your child to get a head start in life. As Sandra Ritter says, "Buy it!" C

April Koppenhaver

Using the VIC 20 With Exceptional Individuals

by Arthur J. Spring, Jr., Ph.D.
St. Mary's College, Winona, Minnesota

Exceptional individuals are, in the educator's jargon, children or adults who are in need of an exceptional effort of will or imagination if they are to develop their human and academic potential. The physically disabled, the mentally retarded, the learning disabled, the blind and the deaf are all considered, by this definition, "exceptional".

The VIC 20 and the Commodore 64 are wonderful instruments for exercising imagination and creativity in efforts to expand the human capacity and the learning capabilities of these exceptional individuals. I have spent some time using the VIC with mentally retarded adults and am extraordinarily pleased with the results. I have also instructed my students in classes in mental retardation on the use of the VIC 20 with exceptional individuals and have received an enthusiastic response.

Most communities in the United States have group homes in which mentally retarded citizens learn to live in a productive community and also develop living skills that foster their own independence. Similarly, most communities have a Developmental Achievement Center, where the mentally retarded can perfect both job skills and basic academic skills related to successful daily living. Both of these institutions exist independently of the work being done for exceptional individuals in all public school districts in the United States.

VIC 20 owners who want to share their knowledge about computing or who wish to apply some basic programming techniques in a setting in which they might be useful, have a ready-made outlet in their communities' special programs for the mentally retarded. One experience of the joy that exceptional citizens have with a computer will be enough to turn a VIC owner into a regular volunteer in work with the exceptional. A telephone call to your local social services agency should be all that is necessary to put you in contact with an administrator at either a group home or an achievement center, where the need for volunteers is always great.

Edouard Seguin, the great pioneer in work with the mentally retarded (whose work inspired Maria Montessori), insisted that the mentally retarded individual be given every opportunity that the normal individual is given. Other workers with the retarded such as Wolf Wolfensberger and Marc Gold have also insisted on developing programs that give exceptional individuals—trained and educated with exceptional outpourings of imagination and determination—the same opportunities any ordinary child would have. The results have been nothing short of astounding. The retarded are now regularly employed in a great many industries, have sheltered workshops of their own and are receiving an increasingly sophisticated academic preparation in our primary and secondary schools.

What can be done with a mentally retarded individual using a VIC? Well, almost anything. The only real limits are the imagination and patience of the teacher. My own work with the VIC in a group home was with exceptional individuals we call the trainable mentally retarded (TMR). These individuals have limited development of academic skills, but are capable of daily living skills and sometimes independent living. The exceptional individuals who comprise this group are a joy to work with and are trusting and cooperative, as well as very excited about computer work.

To work successfully at the VIC with an exceptional individual, one must borrow an insight from Seymour Papert, the developer of LOGO. Papert argues that the computer offers an occasion for interpersonal relationships to develop. The computer, in other words, can be used in the same way as a bike might be used by a parent seeking a close relationship with a child or in the same way as parents might use a stove as a means of getting to know their children better. All of us can use objects as means of furthering our relationships with others, whether we are talking about a book with a friend or the VIC 20 with an exceptional individual.

Secondly, when one is working

with an exceptional individual, one must learn to do what one also does with a friend—talk with questions as well as statements. The first key to working with an exceptional individual is to discover his or her idea of what it is that a computer is or can do. Simple questions about electricity, turning switches on and off and the television set will do much to set both parties at ease and also give the VIC owner an informal assessment of the state of mind of the individual being worked with. It can also reveal the richness of mind of exceptional individuals. As trust and confidence develop over time, the mind of the exceptional individual will reveal more and more of itself.

Some interesting first experiments with the computer are name writing on the monitor screen in different colors and patterns and greeting card manufacture using the VIC's built-in graphics. Most individuals in group homes can at least write their names, and the chance of seeing their names on the TV monitor is something none of them will want to miss once one of their number has experienced that thrill. The VIC owner can initially change the colors of the letters for the student, and then carefully and patiently teach the student to do the changing for herself or himself. This helps the exceptional individual develop coordination and motor skills as well as sequencing ability (the ability to do things in steps). The layperson does not need to know all of the technical terms for the skills the exceptional individual is developing as long as he or she understands that each small step is a triumph of will and understanding. And all of this can be done with the simplest kind of instruction at the keyboard.

To really have fun, and to get a good idea of the wonder resident in the mind of an exceptional individual, the VIC owner can ask the exceptional student if he or she would like to make a greeting card for a holiday on the monitor screen. Valentine's Day is, of course, very easy since the VIC has a ready-made heart symbol. But, with the help of color changes, pumpkins, Christmas trees, Easter eggs, shamrocks and other symbols can quickly be produced from other of the VIC's ready-made graphic symbols. These can then be painted across the screen and alternated with names or printed holiday messages.

Whatever is done, two things should be kept in mind. First, the work at the computer is a means of building human trust and sharing. The VIC owner, even the fuzziest novice, has an opportunity to solidify his or her own knowledge about what can be done with the VIC while getting to know another human being who will repay with enthusiasm every effort made on his or her behalf. And secondly, any work done at the VIC can be a learning experience. If one is "painting" hearts, one can also be counting them. If the exceptional individual can write his or her own name, the teacher's name can also be learned.

Any of the exercises in the computer guide that accompanies the VIC can be adapted to serve as "interactions" with the mentally retarded. The basic rule is plenty-of-encouragement-and-one-step-at-a-time. Color, graphic symbols and simple animation are fun, but even a simple exercise in addition or subtraction can be exciting for both teacher and student. But, remember, whatever else one puts into a program for a mentally retarded individual, one should always be sure to include string variables to represent the name of the individual. The machine, in this way, can actually intensify the exceptional individual's sense of himself or herself as a person whose accomplishment has been recognized.

The excitement one first feels in learning what the VIC can do can be prolonged and intensified by turning the VIC into an instrument for volunteer work in the home community. In working with the mentally retarded, no sophisticated knowledge of programming is necessary—the user guide is enough of a textbook. The VIC's portability and the presence of a TV set in most facilities for the exceptional make the VIC a classroom in itself. One session with the VIC in a group home or achievement center will pay the best dividend of all—an enrichment of one's human understanding and a betterment of one's self through human contact. C

No-Nonsense News About Networking

by Pat Walkington, Gail Austin and Roger Lindholm

When microcomputers first appeared in classrooms across the country, they were purchased in small quantities of one or two systems. These systems were frequently found in media centers or single classrooms with students clustered about.

As schools start to discover the multiuses of microcomputers within the curriculum, such as typing, word processing, problem solving or reinforcing basic skills, administrators and teachers are beginning to realize that whole labs of computer systems are necessary to insure that every student has an opportunity for "hands-on" exposure.

The purchase of many computer systems is difficult not only from a budgetary standpoint (sometimes peripherals such as disk drives and printers cost more than the computers), but from a classroom management standpoint as well. It's difficult enough to maintain one system and one set of software. Picture the teacher who must load sixteen different programs into individual computers while thirty students are waiting!

Because of budget and classroom-management constraints, when it comes to providing a microcomputer lab, many schools are turning to the sharing of resources through networking.

This article will appear in two parts. The first part will deal with what a network is, when you need a network, considerations for classroom layout and the advantages and disadvantages of networking. The second part will outline the networks available for Commodore computers including specifications, cost and availability. It will also include strategies for selecting "networkable" software.

What is a Network?

Networking can mean different things to different people. In this article we will define networking as the sharing of peripheral devices among several computers. Depending upon the networking system used as few as two computers and as many as 16 computers can be linked together at one time. This can be accomplished through the use of special hardware devices or a combination of special hardware and software.

Some of the more simplified networks only allow students to load and save programs. These systems require hardware modifications and are frequently referred to as "passive" networking systems. Other networking, using "active" systems, is much more sophisticated. In addition to the transferring of programs active networks allow the teacher to:

- Control each student's screen.
- Send and receive messages from teacher to student or from student to student.
- Exit a student from the network without disrupting the network.
- Provide a security system to protect the accidental loss of any student programs in the system.
- Access both the printer and disk drive linked to the system.
- Get inside a student's computer to see where mistakes are occurring so that effective individualized instruction can be provided.

When Do You Need a Network? (Don't buy one just to have one!)

You should consider networking if...

1. Your school cannot afford a disk drive for every machine.
2. Datassettes are too slow, or not enough software is available in a tape version.
3. Multiple copies of software create a management problem or a financial burden.
4. You would like more computers available for students.

5. Your teachers want more control over programs accessed by students.

Classroom Layout

When deciding upon a network, layout of the classroom is very important. The safety of the students is the first priority. Does the layout take into consideration:

1. Number and location of electrical outlets.
2. Room for students to move freely about the classroom.
3. Instructor's ease of movement to assist students.
4. Number of students at each station.

The classroom should be laid out so students will not trip over power cords. In addition, the computer lab area should be outfitted with instruction boards that do not use chalk because dust may interfere with the smooth operation of the equipment.

More than two students to a computer leads to a chaotic and ineffective teaching environment. Care should be taken to keep class size to a reasonable limit or rotate students between the computer and lecture areas. To insure appropriate installation of a networking system the Commodore Education dealer in your area should be contacted.

Advantages and Disadvantages of Networking

Networking does not represent a panacea and has some limitations, especially where software is concerned. Software publishers are reluctant to provide single copies of programs for networking purposes because profits are lost. In addition, many programs are not completely networkable. (For example, if a program continually accesses the disk drive, it may not be suitable for networking).

Other problems may occur if programs are improperly loaded into the disk drive, causing the entire system to "freeze" or if poor installation results in unsightly wiring or hazardous conditions for students. Lastly, computers that are networked must be located within close proximity. This may not be desirable if teachers wish to have individual systems in classrooms.

However, if conditions are right, classroom networking can:

- Save money.
- Save staff time.
- Save student time.
- Allow teachers to monitor student progress from a special computer screen designated as the "controller."
- Reduce service and repair costs on extra peripherals.
- Allow students and teachers to operate peripherals more efficiently.
- Reduce the inservice training time required for teachers to learn to load, run and save programs.
- Help outside groups, such as P.T.A.'s, justify the expenditure for computers. Many more computers will be available for student instruction because the cost of the peripherals has been reduced.

SOLUTIONS? Watch the next issue of *Commodore*. C

Educators Learn Computer Literacy at Vermont Resort

Teachers and school administrators can learn about computers the easy way—while they vacation at this beautiful resort on the slopes of Mt. Mansfield.

A well-known ski resort in Vermont has discovered the right ingredients for teaching educators about computers. From the hour they arrive at the Smuggler's Notch Foundation for Computer Use, participants—all of them teachers or school administrators—are totally immersed in a program conducive to both learning—and relaxing.

In contrast to most summer learning programs of this type, this one does not require the educators to leave their families behind. In fact, participants are encouraged to bring their families. Housing accommodations are provided in the resort's condominiums overlooking the surrounding mountain slopes.

Participants attend a five-day series of workshops tailored to their particular computing needs and interests. Workshops are available for educators with little computer experience, educators who have general computer experience but want more and administrative personnel who need to make informed choices about the purchase and use of computer hardware and software in a school district. Credit is available from the University of Vermont.

The Foundation strongly believes that workshop leaders must be working educators actively involved in computer classroom training and teaching. At each level, instruction is provided by these highly skilled and knowledgeable teachers.

The instruction format varies throughout workshop sessions. In-depth presentations are followed by hands-on computer use (the ratio of people to computers is a highly workable two to one). Specific lectures are followed by valuable lab sessions in which participants have direct contact with staff. Groups may meet in the Smuggler's Notch Convention Center or the library to examine new software and become familiar with several hardware models.

The Smuggler's Notch Foundation for Computer Use

1984 Summer Workshop Series Schedule

Date	Workshop	Workshop	Workshop
June 17-22	Lead Teacher	Introductory	XXXX
June 24-29	Lead Teacher	Introductory	XXXX
July 1-6	Lead Teacher	XXXX	Administrators
July 8-13	TERC I-LOGO	Introductory	XXXX
July 15-20	TERC II-Pascal	XXXX	Administrators
July 22-27	Lead Teacher	Introductory	KIPS*
July 29-Aug 3	Lead Teacher	Introductory	KIPS*
Aug 5-10	Lead Teacher	Introductory	XXXX
Aug 12-17	Lead Teacher	Introductory	XXXX

**KIPS—Knowledge Information Processing Systems Workshop*
A workshop for experienced teachers interested in investigating the use of the computer as a tool for thinking and knowledge access and exchange.

INTRUDER ALARM
for VIC 20™ or Commodore 64™

Intruder Interface includes:

1. V100 Alarm Interface
2. Four Magnetic Switches
3. Piercing 95db Siren
4. 100 ft. twisted wire
5. Alarm software

* More accessories available

Total System Only $69.00

plus **$2.75** for postage and handling

Specify Computer and Memory
☐ VIC 20TM ☐ Commodore 64TM
☐ Disk ☐ Cassette

Name _____
Address _____
City _____ State _____ Zip _____
Phone Number ( ) _____
MC or VISA # _____
Expiration Date _____
Send Check, M.O., VISA/MC to:

VIN Systems (U.S.A.) P.O. Box 31
Boone, Iowa 50036

VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd.

YOUR VOICE IN - YOUR VOICE OUT
Digital Recording on C-64/VIC20

Up to 64 numbered words or phrases. Then store as a named file on disk or tape. Words or phrases out in any order from your own BASIC program. New BASIC Commands added. The Voice Master is not needed for response—only for recording. Talking games, clocks, calculators, file data, machine response, advisories—applications too numerous to list. Wherever you want a talking computer with your own natural sounding voice and your own custom vocabulary. Even sing and play music. Many applications in education too. Software for word recognition soon available.

ONLY $89.95

WE CAN DEMONSTRATE OVER THE TELEPHONE!!

COVOX INC.

675-D Conger St. Eugene, OR 97402
Tel: (503) 342-1271, Telex 706017
Check, money order, or VISA/MC
(Add $4.00 Shipping and Handling)

education

Classes may also be held in the Village Barn, a dining/seminar center, where participants meet to share expertise and discuss microcomputer curricula in their local school districts.

After an intense three hours of microcomputer sessions in the morning, participants break for lunch. Lunch may mean a picnic on the mountainside listening to a guest speaker or a discussion among participants and faculty. Following a two-hour instructional time in the afternoon, participants may opt for tennis lessons, swimming, horseback riding or a hike through the Vermont hillsides. As a result of these many informal gatherings participants come away with an expanded awareness of the rapid changes in computer technology and develop strong bonds of group comradeship as well.

Faculty and participants are enthusiastic about the total immersion in computers and the uninterrupted time to learn and practice. Because the Foundation's staff is composed of working educators, it is easy for participants to establish a rapport with their workshop leaders. Instructors dine with participants and share the complexities of computer instruction for classroom use.

This year's workshop format is even more varied than usual, since the Foundation is adding programs developed by the Technical Education Research Center (TERC). TERC, based in Cambridge, Massachusetts, is a nationally recognized non-profit research and development organization dedicated to improving the quality and availability of computer education. The TERC staff will be offering core courses in LOGO in the classroom and Pascal in high school and college instruction.

Foundation President Bradford P. Moore and Program Manager Thomas Petrick have finalized plans for this summer's computer workshop series. If you would like more detailed information, contact The Smuggler's Notch Foundation for Computer Use, Village at Smuggler's Notch, Smuggler's Notch, Vermont 05464, Attention Project Manager.

The Foundation is associated with the University of Vermont; the Vermont Department of Education; Nolan, Norton & Company; the Village at Smuggler's Notch and Technical Education Research Centers. C

The Amazing Adventures of Captain COMAL™:

Captain COMAL Meets the Disk Directory

by Len Lindsay with Captain COMAL

COMAL is an advanced, yet easy-to-use programming language designed to replace BASIC. Here the author of the COMAL Handbook, *along with "Captain COMAL", shows how easily you can use this structured programming language to create application programs.*

Len Lindsay

In this first adventure Captain COMAL shows us how to read a disk's directory from within a running program, sort it if we wish and print it in multiple columns that read like a newspaper. This is a perfect way to print the directories of your disks on labels that you can then stick on each disk, or on paper to cut out and tape onto the disk's envelope.

All you need to use this program is a Commodore 64 computer, a 1541 disk drive (2031 or 4040 will also work if connected with either an Interpod or Buscard), a printer and the COMAL language disk. If you don't have the COMAL disk, yet, it should be available from your local Commodore user group or directly from the COMAL User Group, USA, Limited. Load in COMAL and you're ready to go. Captain COMAL, you're on!

Good day, friends. Today we meet the adorable disk directory, fondly known as $0. She can be seen directly from COMAL when you issue the CAT command. However, we now will undertake to see her from a running COMAL program as well. A little advance planning will help us meet this challenge. Also, COMAL makes it easy to use pieces of existing programs. We will be sharing some, making this a group program.

Here's the plan. First we'll read the disk's directory and store it in an array. Then we'll sort it if needed. Finally we'll print it in neat columns with page breaks if necessary. This may sound hard, but once we break up the job into small pieces it becomes easy. COMAL makes it easy to write a program made up of these small pieces or modules, called PROCedures and FUNCtions.

To read the disk's directory one byte at a time we can use the four DISK'GET routines Steve Kortendick wrote two years ago for PET/CBM COMAL, as modified by David Stidolph for the Commodore 64. To use these routines simply call DISK'GET'INIT once at the start of the program. Then use DISK'GET'STRING to obtain a string of the length specified, or use DISK'GET'SKIP to skip a specific number of bytes (parameters are used to specify this). Now we simply read the directory header, then repeat reading the next file information until the end of the directory is reached:

```
Read the Directory Header
Repeat reading the next file until
the end of the directory
```

Reading the Directory Header can be a separate module. Look at the program listing to see how it is done. Notice in the OPEN statement we call the directory by her familiar name: $0.

Reading the file information also can be a separate module. I called it NEXT'FILE. First it gets the file type (GET'TYPE), then the file name (GET'NAME), and finally the number of blocks the file uses (GET'BLOCKS). Take a look at the GET'NAME module. You already know that you can use up to 16 characters in a file name. But did you know that if you use less than 16, the disk operating system

adds shifted spaces to the end of your file name? GET'NAME changes these into normal spaces so that our SORT routine will be happy.

Another friend of mine wrote the fast SORT routines. Actually, this friend, Borge Christensen, was the founder of COMAL ten years ago! QUICKSORT can be used to sort a string array. It is easy to use QUICKSORT. You don't even have to know how it works. Just include the right parameters: First the name of the array to be sorted, then the number one to start with the first item, then the total number in the array to be sorted, finally the number of characters allowed in each item.

So now the directory is read and sorted. All we have to do is print it. Now is when we can get a little fancy. Look at the procedure PRINT'DIR. It is only 17 lines long, yet allows a variable number of columns as well as variable number of lines per "page". Plus it dates the first page, and titles each continuation page. There are two sections to this module. First two FOR loops and one simple formula cycle through the rows and columns for a page. Then if another page is needed, the starting number of the first name on the next page is set, and then the procedure *calls itself*. This is an example of a recursive procedure. It will keep calling itself until all the names are printed. For easy identification, the title on each page includes the DISK ID.

That's it for today. Look over the program listing. Notice how much easier it is to read than a BASIC program doing the same thing would have been. You may now wonder why you haven't switched to COMAL yet. The first step is the hardest. Once you're using COMAL you won't want to go back to BASIC.

You may have noticed that there are *no line numbers*. Since line numbers are irrelevant to a running COMAL program they do not need to be listed here. COMAL will provide the line numbers for you as you type in the program. Just do this:

```
NEW
AUTO
```

Then enter the lines as listed. Hit the RETURN key twice in a row to stop the automatic line numbering. If you would rather not type in the program it is included on the "COMAL Today Disk #1" from the COMAL Users Group, U.S.A., Limited.

If you would like to contribute programs or modules for others to use send them to Captain COMAL and Friends. LIST the program or module to disk (not SAVE) so it is transportable between versions.

Refer to the COMAL Resources chart if you are interested in using and learning more about COMAL. Captain COMAL will be back next issue. Hope you will be too!

(Captain COMAL is a trademark of COMAL Users Group, U.S.A., Limited)

Commodore 64 COMAL Resource List

Available from COMAL Users Group, U.S.A., Limited, 5501 Groveland Terrace, Madison, WI 53716 (608) 222-4432:

- Commodore 64 COMAL version 0.14 on disk with programs and pocket quick guide: $19.95
- *COMAL Handbook*, by Len Lindsay: $18.95
- Captain COMAL Gets Organized: $19.95
- *COMAL Today* newsletter, 1 year: $14.95
- COMAL Today Disk #1: $14.95
- Turtle Sourcebook, by Jim Muller: $21.95
- Self Addressed Stamped Envelope for Information
- $2 shipping/handling charge per order

Available from Halsted Press, Division of John Wiley and Sons, 605 Third Ave., New York, NY 10158:

- *Structured Programming with COMAL*, by Roy Atherton: $24.95

Available from Educational Company of Ireland Ltd, Ballymount Road, Dublin, 12, Ireland:

- *Foundations in Computer Studies with COMAL*, by John Kelly

Available from Ellis Horwood Limited, Market Cross House, Cooper Street, Chichester, West Sussex PO19 1EB England

- *Beginning COMAL*, by Borge Christensen

COMAL User Groups (complete list in *COMAL Today* newsletter):

- Captain COMAL and Friends, PO Box 6276, Madison, WI 53716
- Alberta COMAL Users Group, Tom Garraway, Division of Educational Research Services, University of Alberta, Edmunton, Alberta, Canada T6G 2G5
- Calgary COMAL Users Group, Robert Allin, 47 Strathcona Place SW, Calgary, Alberta, Canada T3H 1L4
- COMAL Special Interest Group, Commodore Users Group of St. Louis, Richard Schumacher, Box 6653, St. Louis, MO 63125 **C**

Disk Directory Sample Output

```
DEC 25 '83 ID:CO CASE.L              COMAL STARTER    CURSOR0.12.L
ABS.L            CHANGEFILETYPE.L COMAL-VER0.12    DATA.L
AND.L            CHR$.L           COMAL-VER1.02    DELETE.L
APPEND.L         CLEAR'FROM.L     COMAL0.12LOADER  DIM.L
AT.L             CLEAR'LINE.L     COMAL1.02LOADER  DIM2.L
ATN.L            CLEAR'TO.L       COMALERRORS      DISK'COMMAND.L
BASIC'RESET.L    CLEAR'WINDOW.L   COMR             DISK'GET'INIT.L
BOLD'CHAR.L      CLOSE.L          COS.L            DISK'GET'SKIP.L
BOLD'FACE.L      CLOSED.L         CURSOR.L         DISK'GET'STRING.
```

Directory Lister

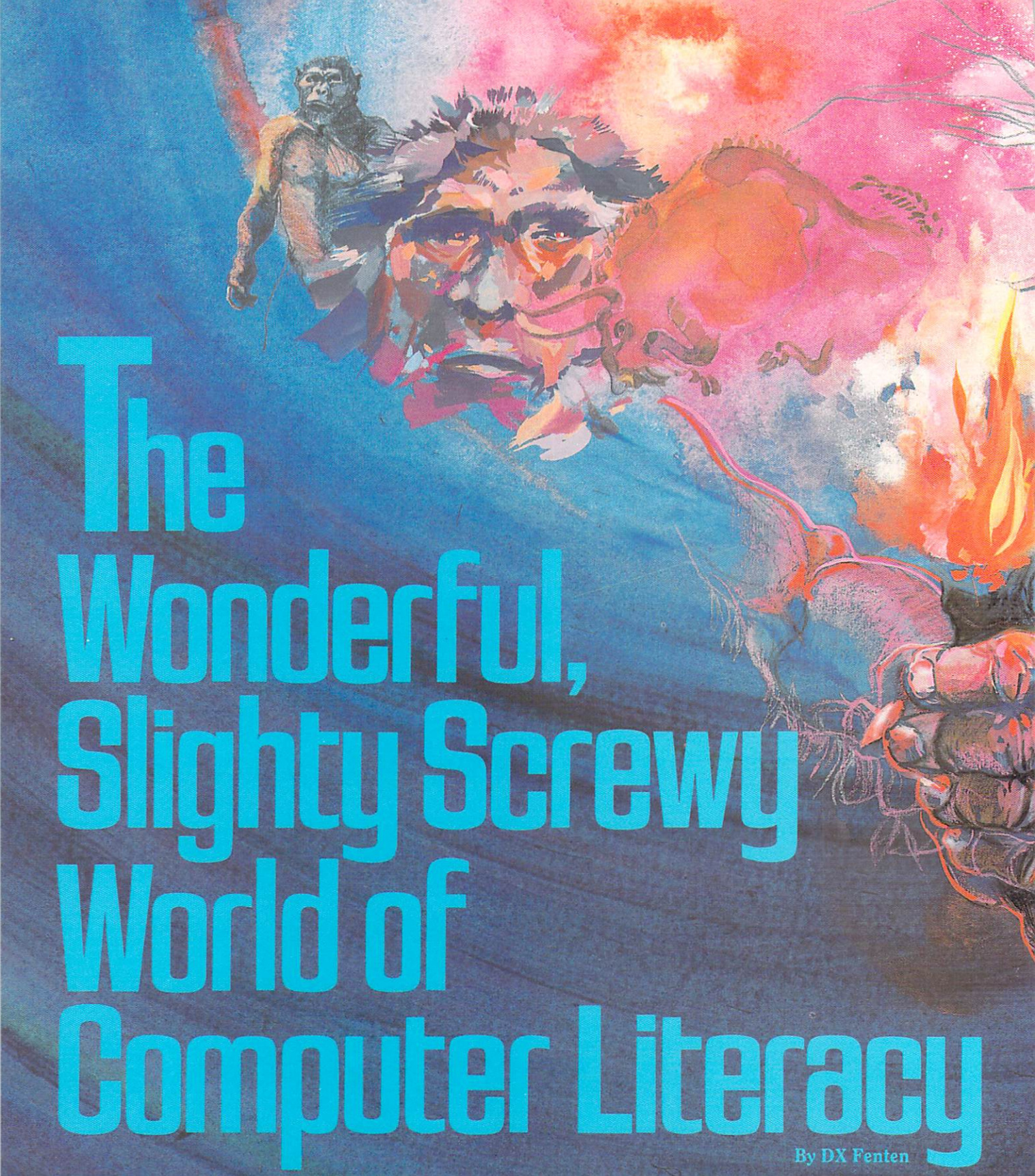
Use your Commodore 64 and COMAL language disk to type and save this program.

```
//delete"0:dir'lister29.1"
//(c) 1983 comal users group (usa)
//list  "0:dir'lister31.1"
// revision date: dec 22 1983
//by Captain Comal and friends:
//Steve Kortendick, David Stidolph, Len Lindsay, Borge Christensen
//
print chr$(147), //clear screen
print "directory lister"
init
repeat
  process'disk
  input "Another disk?": text$
```

```
until text$="N" or text$="n"
end  // all done
//
proc init
  max'files:=144 // max for 1541 disk format
  dim date$ of 10, file'info$ of 40
  dim disk'name$ of 16, disk'id$ of 4
  dim this'file$ of 16, text$ of 1
  dim file'name$(0:max'files) of 16 // array of file names on disk
  dim type$(128:132) of 3 // type of file
  type$(128):="***" //deleted or filler file
  type$(129):="seq"; type$(130):="prg"
  type$(131):="usr"; type$(132):="rel"
  disk'get'init // initialize disk routines
  print chr$(14) //lower case
  set'up'printer //set pitch & lpi
  set'up'run //ask questions
endproc init
//
proc set'up'printer
  // here you can set your printer to condensed mode
  // this will help squeeze more names per line
  // also 8 lines per inch is more compact than 6 lines per inch
  //okidata 92 :: 17 cpi & 8 lpi:
  //print chr$(29)+chr$(27)+"8"
  //mannesmann tally mt 160 :: 20 cpi & 8 lpi:
  //print chr$(27)+"[7w"+chr$(27)+"[4z"
  //c.itoh starwriter :: 15 cpi & 8 lpi:
  //print chr$(27)+chr$(31)+chr$(8)+chr$(27)+"106"
endproc set'up'printer
//
proc set'up'run
  print
  input "How many column listing:": columns
  input "How many lines per page:": lines
  print
  print "Date can be up to 10 characters long"
  print "(ie, 12-22-83 or DEC 22 83 are both OK)"
  input "Date:": date$
  print
endproc set'up'run
//
proc screen'printer
  input "Screen or Printer (S/P):": text$
```

```
  if text$="P" or text$="p" then select output "lp:" // printer
endproc screen'printer
//
proc process'disk
  file'count:=0 // initialize count
  input "Insert disk in drive 0:": text$; //; means stay on same line
  print "OK"
  print
  read'dir
  sort'dir
  screen'printer
  print'dir(file'name$,1,columns,lines)
  select output "ds:" //screen output
  pass "i0" //reset bus
endproc process'disk
//
proc read'dir
  read'dir'header
  block'count:=0 // for 8 block cycle
  blocks'free:=664 // empty 1541 disk free blocks
  repeat
    next'file
  until file'end
  close
  print "blocks free:";blocks'free
endproc read'dir
//
proc read'dir'header
  pass "i0" // initialize drive
  master'file:=2
  file'end:=false
  open file master'file,"$0",read
  disk'get'skip(142,master'file,file'end)
  disk'get'string(disk'name$,16,master'file,file'end)
  disk'get'skip(2,master'file,file'end)
  disk'get'string(disk'id$,2,master'file,file'end)
  print "Disk:";disk'name$;"ID:";disk'id$
  print
  print "num file name        typ blocks"
  print "--- ---------------- --- ------"
  disk'get'skip(92,master'file,file'end)
endproc read'dir'header
//
proc next'file
```

(continued on page 64)



The Wonderful, Slighty Screwy World of Computer Literacy

By DX Fenten

Wherein syndicated columnist DX Fenten attempts to answer the sticky question, "What is computer literacy and why must everyone have it?"—and raises a few additional questions in the process.

Illustration—Robert Neumann

You've certainly heard it enough times, and so have we—"In today's world you must be computer literate." Or, put another, even more familiar way, "Computer literacy is a must for anyone who hopes to get anywhere in today's world." This kind of thinking is permeating schools, businesses and homes. It has become so much a part of the idiom by which we now exist, there just seems to be no further doubt about it... anybody who wants to be somebody had better have computer literacy. Making that statement is easy and not open to much dispute since "everybody's saying it." The hard part is answering the question, "What is computer literacy, and why must everyone have it?"

Over the past few years, the key words in education, as in most fields, have been (you guessed it), "computer literacy." Though there seem to have been trends in education that lasted about five years and were then relegated to the dusty corners of storage closets, the movement to computers in education seems as if it will be a permanent one. This should answer, at least in part, the question of what young people are learning in school that will be of practical value later on in their lives.

In a survey conducted by *Electronic Learning* magazine in 1981, they found that just about every state in the union was sitting on its collective hands, waiting to see if computers "were really here to stay." Educators seemed to feel that, this time, they had to be certain before they plunged ahead and got involved in training for both teachers and children. Minnesota was the only state listed in the survey as heavily involved in promoting the instructional use of computers. Florida had passed a legislative mandate saying it was state policy to use computers in instruction, but did little about that mandate.

A new survey conducted late last year still shows a "wait and see" attitude on the part of most states, but at least now there are considerable signs of getting with the times. According to this survey, at present six states require their schools to teach their students computer skills and 12 states officially recommend it. A grand total of 47 states "have launched campaigns aimed at enabling their schools to impart those skills to their students."

Just as important, two years

ago there was only fragmented support for teacher in-service and other computer training courses. Today, says the survey, the number of state agencies "actively involved in computer in-service projects has climbed to 40."

One of the most comprehensive computer literacy curriculums is in use in Cupertino, California. The entire concept—which was written for K-8 but which also makes provision for additional teaching in the secondary schools—has as its foundation the premise that all students should "have an opportunity to become computer literate." Computer literacy, the Cupertino schools believe, "is the ability to function in a computer- and technology-oriented society. Students will understand computers and their applications and implication in the world around them. They will develop the skills necessary to communicate with computers and recognize the computer's capabilities and limitations."

The teaching starts in kindergarten and is reinforced and expanded as the student moves through the school. Included in the program's Computer Awareness section are recognition of the makeup of a computer (identify parts, run a program, list different languages), description of how computers affect our lives (common uses, occupations, value of computers for future employment), history and moral issue of computers and much more. In the programming section, the student is required to perform basic programming skills, perform specific skills and do simple tasks using LOGO, Pilot and BASIC languages.

Rounding out the curriculum, students are required to use computers, or at least understand how they can be used, in such areas as social studies, language arts, science and math. There's little doubt that a program of this type, if implemented in schools throughout the country, would quickly eliminate any possibility that either Johnny or Jane could ever be computer illiterate.

The man who seems to have coined the term "computer literacy" some ten or twelve years ago, Arthur Luehrmann, makes a very strong case for programming as part of computer literacy.

It is quite obvious that programs of this type and scope (as well as anything else that's taught) require teachers who have acquired the knowledge and skill and are motivated to teach the subject. In far too many school districts, however, teachers in any discipline other than math are, to a great extent, allowed to sink or swim on their own when it comes to computer literacy.

In many districts there is a supervisor or coordinator (declared to be an "expert" by the administration) who gives some "hands-on training" to those teachers who volunteer. The teachers are taught how to LIST commercial programs, find out "how to make the computer go," and learn some very, very basic BASIC programming. They are then turned loose, with a sheaf of notes and suggestions from the supervisor, on their classes. Meanwhile, the other, non-volunteer, teachers continue along their merry way, allowing the computer, sitting in the corner of the classroom, resource room or media center, to gather dust.

However, fortunately, many other school districts, through their administrators, are now more computer aware, if not completely computer literate, and are starting to provide the training required to help teachers conquer their fear and their misunderstandings of computers.

It is important to note that just about everything cited for education can be cited similarly for business. There is, in many companies, token teaching, token training and token acceptance of the concept that people comfortable with computers are a great asset to the company and to themselves. The people who learn to use computers (mainly on their own) are praised and the others are left to their own devices.

To hear some computer users in both business and education tell it, computer literacy simply entails being able to use a computer, not "being afraid" of it, and knowing reasonably well what it can do for you. These people rarely mention programming in their definition, and if there is any mention, it's in the form of a nice, but optional, "extra added attraction." They consider a computer a kind of a simple machine or tool, much the same as an electric typewriter or calculator.

There is some truth to this. You may, for instance, have a stereo you enjoy tremendously, but you need not know how to write music nor play any instruments to enjoy this piece of equipment. Even learning to choose your own components or building your own system is unnecessary, since perfectly wonderful systems can be purchased as a unit, taken home, plugged in and enjoyed by any and all.

In the same manner, if you are able to use some of the excellent software that is now so easily available, software that has been created and designed by experts in specific fields, to get your computer to do what you want it to do, why should you be considered illiterate if you cannot also program that computer? There is also a reasonable argument put forward by "non-programmers" that there is

no sense in learning programming today and having it hopelessly out of date tomorrow. They say that since computer designers constantly make the machines easier to use by incorporating more processes right into the units, the need for even minimal programming seems to be dissipating.

On the other hand, for some organizations even knowing programming is not enough to be considered computer literate. A recent report by the Academic Council Committee on Computers and Information Technology of Stanford University lists three main criteria for computer literacy: 1. some facility with at least one text editor, computer and operating system; 2. the ability to write and debug a program successfully; and 3. the ability to design, implement, debug and maintain reliable algorithms, perhaps even efficient ones, in the service of serious professional goals.

How many people successfully using computers today can honestly even begin to consider themselves computer literate based on these criteria? How many of these people would you consider computer literate—how many illiterate? How many of the people using computers successfully on an everyday basis could give you a good definition for algorithm? How many would even try and how many would shrug their shoulders and say something like, "What difference does it make?"

The man who seems to have coined the term "computer literacy" some ten or twelve years ago, Arthur Luehrmann, makes a very strong case for programming as part of computer literacy in his book by the same name. He suggests that a computer is what a computer does, so you should be able to make a computer work in order to understand it.

But which computer are we talking about and, once that has been decided, which programming language should be taught? If a person is computer literate on one computer and uses one programming language on that computer, is that person truly computer literate? Should the criteria for computer literacy be "softened" a little bit? Does it, in the light of this multiple-computer, multiple-language world, make more sense to suggest that people should know how to use computers, should be able to find and use the programs that will cause computers to do what they want them to do and then, if at all possible, be able to do some programming to make all of this even more exciting, even more valuable?

Regarding the idea that mastering different computers is difficult and time consuming, it is important to point out, and programmers will quickly concur, that a computer language is a computer language—that BASIC is BASIC—no matter the computer. If you can program on one computer using BASIC, it would only be a matter of reasonable time before you could also program on another computer with that same language, even though the BASIC used on one computer could be different from that used on another computer.

If computer literacy has been on your mind and you feel there is a great need, for any one of many different reasons, to become computer literate, it's vital that you decide, before you go very much further, the level you feel you want and need to achieve. Are you interested in using the computer as a pure piece of equipment? Combining it with software, making it do the thousands of things it can do, with no more involvement on your part than to put in some information and take out the results? No analysis, no additional information than that put in and synthesized? You could be computer literate doing this, but you'd certainly be missing a lot, too.

Would you prefer using the computer as a tool that would help you, by providing analyses and additional information, make the decisions that are required in your schoolwork, in your business, in your life? If you are really comfortable working on a computer, have selected the proper software and are able to understand and perhaps even do a little programming, you'll be really getting your money's worth out of this even higher level of computer literacy.

At this point, let's add a few words about the creative side of computers and computing, even if programming never becomes your "thing." If you want to do some writing (perhaps even about computer literacy), your creativity can still rise, shine and run rampant with the assistance of your handy dandy computer word processor. Then, perhaps, just perhaps, someday you'll be tempted to learn some programming, some creative programming—the kind that really makes the computer "do its thing"—and you'll discover the joy and pleasure that comes from being computer literate and using a computer for what it does best.

This then is the true "key" to computer literacy. Learning about a computer, becoming fascinated enough to want to learn more about computers and computing, and then realizing that learning some programming completes the picture and allows you the full pleasure and accomplishment that can come from this wonderful piece of modern technology. And with it, the thought, hasn't it always been that way...? C

DX Fenten is author of the nationally syndicated column "Computer Bits". He is also Garden Editor for Newsday, *a Long Island, New York, newspaper.*

Get Lite

Resources for Computer Literacy

By Diane LeBold
Editor

Once you decide approximately WHAT you need to know about computers, the question becomes HOW do you learn it? Here are several common alternatives for gaining greater understanding of this versatile electronic tool.

rate

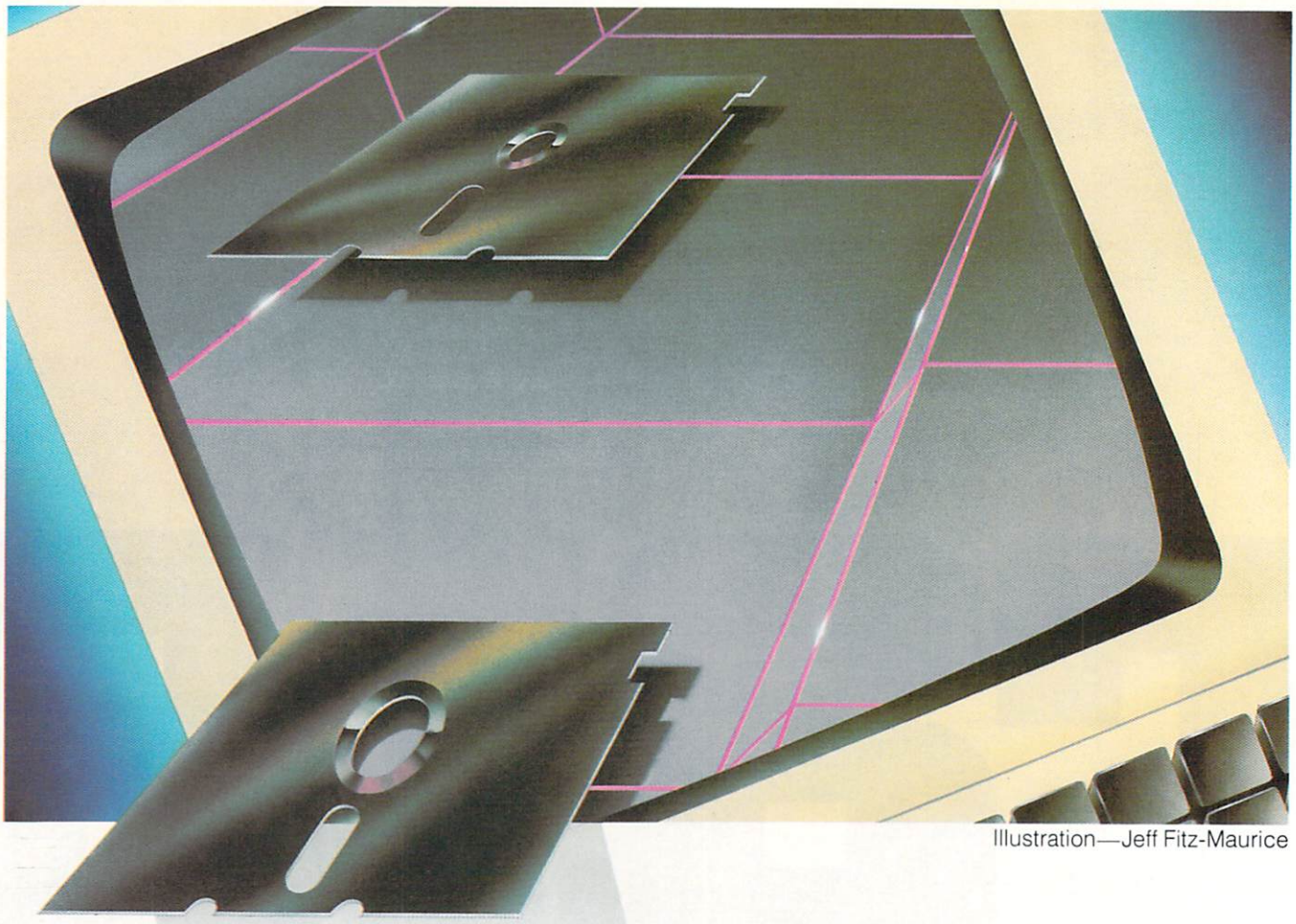
We know many of you are avid programmers who would rather debug that data base you've been working on than eat or sleep and that others of you are hardware hackers who love to fiddle around with the electronic innards of your equipment. But we also know that many of you are not at all interested in programming or electronics—you want to simply sit down and use your computer without getting into the nuts and bolts of it all.

The question inevitably comes up when you're talking about "computer literacy"—which of these is the "correct" approach to microcomputing? Does a person have to be a programmer or a hardware techie type to be truly "literate"? Let me ask you this. Do you have to be an automotive engineer to drive and properly maintain a car—to be "car literate"? No. But, on the other hand, it's pretty evident that the more you know about your car, the better off you are, right? (Many additional analogies spring to mind but you get the idea.)

What fascinates me about the new literacy—and microcomputing in general—is that it hasn't yet been institutionalized. No one has decided, yet, exactly what is "correct" and "incorrect" when it comes to learning about microcomputers. There is no official curriculum for achieving microcomputer literacy and few "sanctioned" credentials for teaching it. As far as I know, no universities yet offer a degree specifically in microcomputing and, in fact, many well-known micro experts are self-taught. Most people, it seems, from beginners to experts, are more or less playing it by ear—ad libbing according to their own interests and experience.

I suppose many people are distressed by such a haphazard approach to learning. Personally, I'm excited by it because it means people can pursue computer literacy in many different ways from any of several different angles, depending on their particular interests and needs. This, it seems to me, is how real learning happens. And any time people *really learn* something new, whether it's computers or zoology, I think it enhances their lives.

Let's take a look, then, at some of the methods people are using to learn about computers. Some methods provide only an elementary understanding of the most basic facts of computing, some get you intensely involved with programming and/or hardware and most are somewhere in between. The method you choose (or have chosen) depends entirely on your personal interests and inclinations.



Illustration—Jeff Fitz-Maurice

Teach-Yourself Packages

If you're the kind of person who learns well on your own and has a fair amount of self discipline, you stand a good chance of learning a lot about computers from teach-yourself packages. These often consist of a book and cassette tape or disk. Commodore's own *Introduction to BASIC* for both the VIC 20 and Commodore 64, for instance, offers a complete course in basic BASIC programming. It contains a book that explains various programming techniques and a tape that lets you try the techniques on your own to see exactly how they work. Other similar packages are available from companies like Creative Software *(I am the C-64)* and Comprehensive Software Support *(C64 Tutor)* for reasonable prices. Even if you never become a great programmer, you'll definitely gain a greater understanding of your computer's capabilities using these kinds of simple teach-yourself packages.

Other teach-yourself packages might be more general, providing an overview of what your computer can do without getting into the nitty gritty of actually writing programs. World-renowned Commodore expert Jim Butterfield, for instance, has created a Commodore 64 video training tape that explains how to get started using your Commodore 64. The video is aimed at making beginners comfortable with their computer, and is available for $39.95 from PF Communications, 2727 N. Grove Industrial Drive #101, Fresno, CA 93727.

A somewhat similar but much more superficial video explaining the 64 is also available from Hayden Publishing, although that one is recommended only for people who know very little and don't intend to learn much more about computers.

Although it's not exactly a teach-yourself "package"—in that you can't go out and buy it like you can the others—it's worth mentioning here that Butterfield has also created a series of computer literacy television programs titled "Bits and Bytes" that are now being televised in the United States via the Public Broadcasting Service. These half-hour shows, produced by TV Ontario, were well-received in Canada last year and provide an enormous amount of solid information about computing. They're aimed at beginners, but even people with a fair amount of expertise have found them interesting and informative. Watch for them on your local PBS station.

School Systems

Many children are getting their computer literacy right along with their three R's (who was it suggested a new three R's—READY, RUN, RETURN?). If your children aren't being taught computing, I'd suggest you find out why. If they are, I'd suggest checking into the curriculum to find out exactly what they're learning—and who's

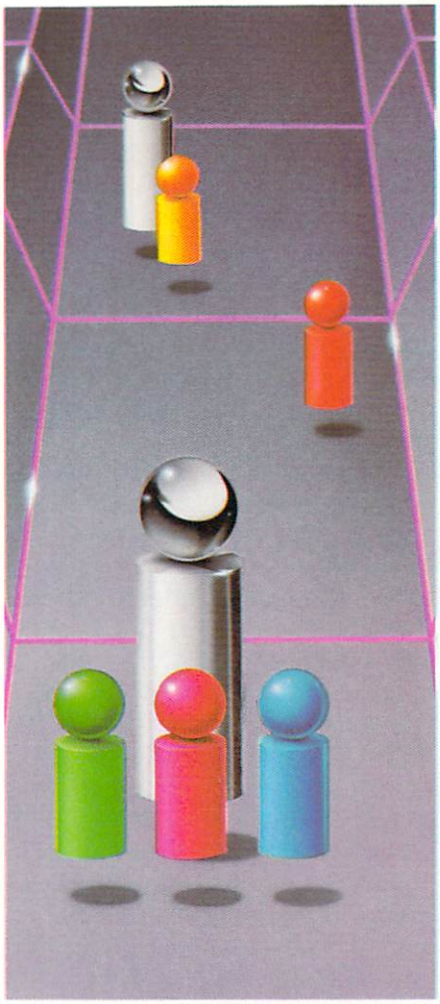
teaching them. Remember, some children think they're learning how to use computers when all they're really doing is running a computer program that helps them learn math or English. There's a big difference.

Since DX Fenten has provided some insight into what school children are learning about computers (page 35) I can skip right on to another service many schools now provide—adult night classes in computing. These classes vary in their content and scope, from the most rudimentary introduction to computers right up to complex programming, but regardless of what level they are aimed at, make sure they actually *have computers* to work on. Just learning theory can be more confusing than learning nothing, sometimes. If your district doesn't offer these kinds of courses and you would like to see them instituted, try contacting your school board. If they say they need teachers, try contacting your local user group to get someone to teach. If you don't have a local user group—start one. Which brings us to:

User Groups

There is no doubt in my mind that user groups are the single most important force right now in helping people get literate. As of this issue we have over 400 Commodore user groups on our list (see page 116) and I'm pretty sure many more exist who just haven't let us know. I know many of our groups have memberships in the hundreds and a few (the Calgary, Alberta, group, for instance) have a thousand or more. Of course, the Toronto Pet User Group (don't let the name deceive you—they cover all Commodore computers) is the granddaddy of them all in both longevity and size. It was one of the—if not THE—original user groups in North America and now boasts about 14,000 members, last I heard.

A good user group's sole function is to help its members learn more about—and get more use out of—their computers. Period. If it's doing anything else, don't bother to join. Generally the groups provide four basic services: regular meetings, a newsletter, a library of books and software and special events like demonstrations of new equipment, seminars or local computer "fairs". Some run telecommunications bulletin boards, as well, and some of the larger groups sponsor computer camps and courses at local colleges and schools. Some groups are sponsored by or affiliated with a computer dealer, college or high-tech company.



Some groups are formal and hold regular parliamentary-style meetings, followed by a speaker or demonstration. Others are more casual and just let people mingle and discuss whatever comes up.

There are as many variations on meeting styles as there are groups. The one thing they all have in common, however, is that their members learn about their computers, find out about new products, see software and hardware demonstrations, get a chance to discuss problems and ask questions, pick up programming and other technical tips, have access to books and software—and meet new people.

As you might suspect, there seem to be no age barriers among members of user groups. Most groups have youngsters, oldsters and every age group in between, all bound by their common interest in computing. It's amazing to me that computers, once dreaded by some as a potentially destructive and divisive force in our society, should turn out to be a catalyst in many people's social, as well as intellectual, growth.

If there are no user groups in your area, you might consider starting one. With over three million Commodore computers out there, regardless of where you are, it's not too hard to find other Commodore users who are interested in participating. If you want information on how to start a group, I'd suggest contacting an already established group in your state to find out firsthand how they went about it. We also occasionally run "User Group Spotlights" in *Commodore* and *Power/Play* magazines, where you can pick up some hints on how to start a group and keep it running. Just be sure, before you plunge into starting a group, that you have enough time to devote to it—or have enough friends to spread the work around—because it can take a fair amount of time to keep things going, especially if the group starts to get large.

Telecommunications Networks

Ever since Commodore began marketing inexpensive modems (the device that translates computer signals into telephone signals and then back to computer signals at the other end) an astounding number of people have begun using their computers to communicate over the telephone lines. You can find quite a selection of information by tapping into the

various data bases and bulletin boards available via your telephone—and at least a portion of this information is devoted specifically to helping you learn more about computing.

The most obvious telecommunications network to start with is the Commodore Information Network, available via the CompuServe Information Service. (When you buy a modem for your VIC 20 or Commodore 64, you get a free subscription to CompuServe and instructions on how to access information.) Here you'll find a hotline, where you can have questions answered within a day or two by our trusty SYSOPs (systems operators), product announcements, abstracts of articles from our magazines, software and hardware tips, directories of user groups and dealers, and bulletin boards where you can "meet" other Commodore owners. You can also hold private conferences with other users on the Network, or "attend" a Commodore-sponsored conference on a specific topic, usually featuring an expert or two.

Once you get the hang of how to get into the various data bases and bulletin boards, you'll discover a wealth of information on computing (as well as a whole bunch of other topics). Try checking *The Computer Phone Book* by Mike Cane, published by Plume, for information on networks, bulletin boards and SIGs.

If you have a modem, then, you can add to your "literacy quotient" quite a bit by snooping around in these various networks. You can also find out about stocks, make airline reservations, access encyclopedias and even attend a telecommunications university for credit—but that's another story.

Computer Camps

I know of at least half a dozen children's summer camps that now offer instruction in computing and I assume there are many, many more. You can often find their ads in the back pages of recreation-oriented magazines or in computer magazines aimed at kids. Another source is *The Computer Camp Book*, available for $12.95 plus $2.00 postage from: The Computer Camp Book, P.O. Box 292, Yellow Springs, OH 45387.

If you're considering one of these camps as a serious method for helping your kids get literate, check on the ratio of computers to children. Obviously, if there are two computers and a hundred kids nobody's going to get much hands-on experience.

If you're interested in adult camps, you might check with a local college. Lincoln College in Lincoln, Illinois, for instance, has had great success running a summer Commodore computer camp for older kids and adults. People stay in the college dormitories and attend classes taught by expert faculty like Dick Immers and Jim Butterfield. For about five days everybody is completely immersed in Commodore computers. Once in a while they eat, I think. For more details on that camp, consult *Commodore* magazine, Issue 27.

The camp at Lincoln College is aimed at users with some knowledge of computers, but you can obviously tailor programs to suit any level, from beginner to expert. If your local college or junior college would be interested in running this kind of computer camp, I'd suggest they contact Robert Widmer, at Lincoln College, Lincoln, Illinois 62656, to get some hints on how to set up a successful program. If they already have a summer program, by all means look into it because any kind of exposure, regardless of how you intend to use your computer, ultimately helps you understand your versatile new tool better.

For those of you who are involved in education, by the way, I know of at least one summer camp aimed specifically at teaching educators about computers. It's conducted by the Smuggler's Notch Foundation for Computer Use, Inc. in Smuggler's Notch, Vermont. (See our Education department, this issue.) Throughout the summer the foundation offers five-day workshops, some for administrators, some for teachers with major responsibility for teaching computers and some for educators with no prior computer experience. With some additional study and a written report, participants can qualify for three graduate credits at the University of Vermont. You can write to the foundation at The Village at Smuggler's Notch, Smuggler's Notch, Vermont 05464.

Professional Computer Schools

Here we have a rapidly expanding growth industry if I ever saw one. Computer schools, it seems, are springing up all over the place like buttercups in May. Unlike buttercups, however, many of these schools are not rooted in any one location. They're portable institutions, carrying their equipment with them from place to place, setting up, running a four-week or eight-week or twelve-week course, packing up and moving on to the next job.

Northwest Educational Associates' "Have 64 Will Travel" program, for instance, seems pretty typical. Run by Anne Winchester, former Dean of Women at Washington State University, and former teacher Darlene Atteberry, the Seattle-based school has ten Commodore 64's, which are networked together in the classroom —wherever that classroom might be. They began, according to Anne, by teaching courses at a local church. Then they started teaching at other locations for the city parks department, and now are developing other clients as well.

The two women are concerned primarily with reaching people who normally wouldn't have an opportunity to learn about computers. They emphasize home-oriented applications and teach many aspects of computing from how to buy a computer to evaluating and using software, right up through programming, depending on the need. If you're interested in contacting Anne and Darlene

(who, by the way, have an article in this issue on page 45), you can call 206-722-3002.

Other similar itinerant schools can also be found throughout the country. Futureschool, for instance, headquartered in Madison, Connecticut, operates out of several different locations, many of them YMCA's, and has programs for both children and adults. Louis Esposito, president of Futureschool, sees the school as an "educational package" rather than a physical school—a package that can be set up anywhere.

Futureschool differs from other computer schools in that it uses a distinctive teaching method called the ZuseMethod, developed by the school's founder, DeWitt Zuse. It is similar to other schools, however, in its philosophy that people who can't control computers are at the mercy of those who can. You can contact Futureschool at P.O. Box 1129, Madison, Connecticut 06443, 203-453-2577.

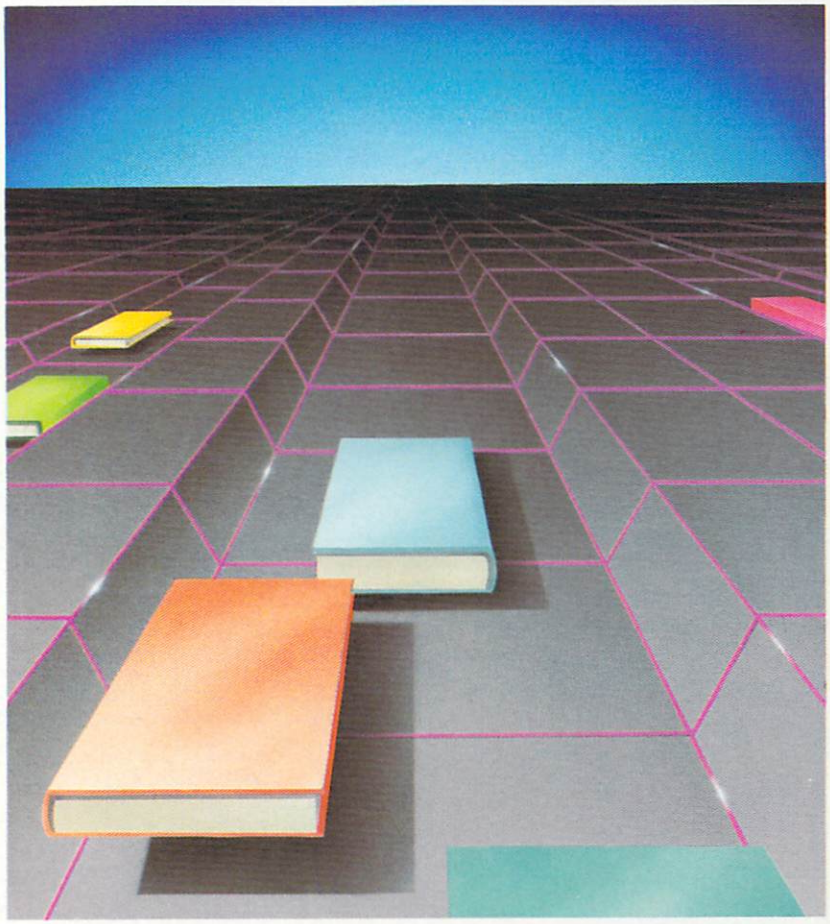
If you'd like to gain some insights into your computer by attending a professionally run school like these, check your yellow pages under "Schools" or call your local user group, YMCA, recreation department, community center or community college for information. If there don't seem to be any such schools in your area, this might be your opportunity to set out on an exciting entrepreneurial adventure and start one yourself.

Books and Magazines

I've saved this one for last because Jim Strasma's article on page 57 covers books in detail, and I just wanted to sneak in a few comments of my own here at the end.

Obviously, reading books is a good way to learn about computers, although not all books are entirely accurate and some are even misleading. You can sort out the good from the bad in several ways. You can consult a knowledgeable friend or someone from your local user group, check your friendly *Commodore* magazine (or any of several Commodore-oriented magazines) for reviews or simply go on instinct.

Two national bookstore chains—Waldenbooks and B. Dalton—carry a full line of computer books, covering just about any facet of computing you might care to examine. You'll find very general books, like *How to Get Intimate With Your Computer* by Fred D'Ignazio, very simple programming texts like *I Speak BASIC to My...* (fill in PET, VIC or Commodore 64 at the end of that title) by Aubrey B. Jones, Jr., right up through *Machine Language for Beginners* by Richard Mansfield and the various programmer's reference guides produced by Commodore, which are intended for more sophisticated programmers.



Finally, a grand way to increase your understanding of computers is right here in front of you—user magazines. In addition to Commodore's own magazines, *Commodore* and *Power/Play*, there are four other magazines that I know of devoted strictly to Commodore computers, and this doesn't even take into account the myriad of general-interest computing magazines, most of which are at least somewhat helpful, although I'm inclined to think that magazines devoted to specific computers are probably the most useful.

Try computer dealers, B. Dalton and Waldenbooks to find these magazines. If they don't carry what you want, ask them to order it. If they can't order it, call the publisher and let them know.

Finally

How many adults do you know who can't drive a car? Who are at a total loss around a typewriter? Probably not many, right? In fact, people who can't drive and/or type can be at a great disadvantage in this society. We're quickly moving toward a time when the same will be true of computers. That's why learning about computing—getting literate—is important, right now. And besides, computers are great fun.

Why miss out? Get out there and get literate—or, if you've already achieved that glorious goal, get off your duff and help somebody else do it! C

Computer Literacy:

Different Things to Different Users

By E. Anne Winchester and
Darlene K. Atteberry

Who is seeking to become computer literate? What do they want to know about computers? How can you teach it to them? These questions are answered by two educators who teach computing to all kinds of students, from youngsters right on up through public school teachers.

The kids enter the room and excitedly run over to the nearest computer. A bank of ten Commodore 64 computers has been moved into the local community center to teach people how to use home microcomputers. Most of the children have no fear of computers and immediately feel at home, having played video arcade or home computer games. They start to shout and carry on about what to do to make the computer work. Their attitudes are positive, but their computer literacy is limited to their familiarity with video arcade games. Some children express apprehension but, with few preconceived notions about computers, they forge on with eagerness to discover what a computer can do.

The next class of the day yields a group of women who express concerns that their children are using computers in school and their husbands are working on computers at work. There is a definite communication gap at the dinner table. Feeling left out, these women are seeking to become computer literate. Their level of expectation is mixed and includes computer anxiety as well as excitement about learning something new.

Illustration—Greg Purdon

Classes organized for teachers bring a group of people who are eager to learn but who often have a great deal of trepidation and fear about what computers might mean for their future.

The women share some common characteristics. Most have education beyond high school and either work part-time or desire to go to work in the future. They know that in order to survive in the work world that they will need to know something about the computer. Some contemplate having some type of business in their homes using accounting, word-processing and other business applications. Some have heard that a good use of the computer would be to keep track of their home recipes, address books and club memberships. Other women in the class are just plain curious.

At night the adult and high school class brings people who work or go to school during the day. The adults are aware of or use fairly complex business computers at work and want to learn to use home computers. The students either can't fit a computer class into their schedules or go to a school that has no such classes. Most of these people are under the impression that it is necessary for them to become computer programmers to make the most effective use of their computers. They are aware of software packages, but are not clear about how computers can be used to assist them in the everyday work world.

Classes organized for teachers bring a group of people who are eager to learn but who often have a great deal of trepidation and fear about what computers might mean for their future. Will they become familiar enough with a computer to use it with ease? Will they be able to keep up with the students who seem to be ahead in learning about computers? Will the computer become an adjunct teacher and assist in teaching, thereby enriching the classroom experience? Their knowledge about computers is limited because they lack exposure in their work environment. While many people in the business world use computers in everyday life, teachers generally have not had such exposure. The educational community uses computers for administrative functions, but use of the computer in the classroom is just now "becoming".

While it is convenient to classify people in these kinds of broad groups, each group has its stars, slow learners and variations. In each case, the attitude that the person holds toward the computer sets the tone for their learning experience. If they are excited and eager to learn, they move forward no matter what problems arise. If they are fearful and anxious, it takes time to win them over to the "friendly" computer sitting in front of them.

The kids want to see their computer monitor go "Pop, Bang, Smash, and Fizzle." The "Women Only" class wants to be sure that if they touch the machine, it will *not* go "Pop, Bang, Smash or Fizzle." The teachers know they need to conquer this strange mechanical contraption before it takes over the classroom and the evening group thinks they are about to become computer programmers. Each person has a different level of knowledge and each person has a different set of expectations.

Fortunately, instructors, too, have some ideas about computer literacy. Each instructor has been through classes where the computer was not present while they were being taught—usually a series of history lessons about where computers came from—and has listened (not too intently) to lectures about the technical insides of the little machines. All the time they were thinking, "But where is the computer? How can I learn about computers without touching them, without trying a FOR/NEXT loop or experiencing some of the horrors I have heard about in these classes?"

It is as if it were necessary to have a full technical knowledge of the car before one can be licensed to drive. Or to hear of the history of the automobile from Henry Ford's day to present to be able to appreciate the machinery. It is possible, even as pervasive as the automobile is in our society today, that many of us will never know how it all works. Instead, what people want to do is to get into the car, turn on the engine and be able to smoothly and efficiently operate the car.

The same is true with computers. People want to be able to use them, not just talk about them. There is no better analogy than the car, because the computer will be as pervasive in the future as the car has been over the recent past. To attend a class to learn to drive and be faced with spark-plugs, compression ratios and axle length would not provide a person with the total picture of what a car really is, nor would it achieve the goal of the person in learning to drive.

Computer literacy should denote competency in operating a computer. Addressing the needs

The kids want to see their computer monitor go "Pop, Bang, Smash, and Fizzle." The "Women Only" class wants to be sure that if they touch the machine, it will *not* go "Pop, Bang, Smash, or Fizzle."

of the person and the defined goals in learning about computers should drive the class. Having a computer to work on, touch, make mistakes on, and just "play on" offers the beginnings of computer literacy. Can you make the computer do what you want it to do? Does it accomplish what you need to achieve?

Literacy in the true sense of the word means to be educated about something. To be able to read and write, for example, is a prerequisite to being educated. The same need is apparent in being computer literate. One must speak the language and be able to communicate in that language. To a person pursuing a computer programming career, that might mean learning BASIC, Fortran, COBOL or Forth. However, if you again address the needs of the majority of users, you will find that learning essential commands to operate software packages and being able to interpret documentation will allow most people to achieve an appropriate level of computer literacy.

In the kids' classes, literacy can be handled in a fun, game-oriented manner. This is where the Commodore 64 really shines. They learn where the keys are on the keyboard through the use of typing tutorials, which simulate the experiences they have had with arcade games. When they are learning programming the kids can immediately be introduced to color and how to access the various color modes. They "draw" with color, create pictures, learn to use graphics and build programs with color combinations that enhance their beginning skills. LOGO can be introduced in one session to illustrate another type of programming language. Use of the disk drive and loading public domain software teach the children to respond to pre-programmed directions as well as giving them opportunities to play games, create music and solve problems. Time should be left for playing because children learn through experimenting.

The "Women Only" and adult evening classes we teach offer a diversified range of knowledge about the computer. Here we use special programming to teach where the keys are and how the computer operates. Class members are taught how to use a disk drive and printer. An electronic spreadsheet is used and a class project is developed that creates a home budget, inventory system or other applications suggested by the group. Using a word processing program, class participants create a document, and finally, learn to enter information in a data base system.

In our teachers' classes, literacy takes on special meaning. Teachers must be computer literate because many will soon be teaching computer literacy to students. Learning how to operate networking systems, finding a suitable word processing system for various ages of children, learning to make up tests or to keep grades in a computer grade book are emphasized. Again, teachers are given time to experiment and make mistakes while in an environment where someone can help them understand why error messages appear.

These classes constitute beginning learning experiences or a survey of what the computer can accomplish for the individual. It is at this point that most people realize they must select from a broad continuum of functions that a computer offers. Their goals begin to form in their mind and they chart a course to accomplish those goals. Are they computer literate or have they just begun? The answer lies with the individual. To understand what a computer can do may have been enough for them or they may wish to advance to a particular application such as word processing, mail lists, check book balancing, inventory control or programming and will proceed with their computer literacy. Education can be a lifelong learning experience. Computers and the multiple uses of computers may also be a lifelong learning program and an exciting one at that. C

Anne Winchester (l.) and Darlene Atteberry run Northwest Educational Associates, an independent computer education company based in Seattle, Washington. They teach computer workshops and provide computer consultation services throughout the Seattle area, and can be reached at 3114 37th Place South, Seattle, WA 98144, phone 206-722-3002.

INSTALLING YOUR COMPUTER

Of all the areas of computer literacy, the one most often overlooked is the computer operating environment. Things like furniture, flooring, power supply and telephone lines are often remembered

Where is it Going to Go?

The first step is to size up the room or area that the computer system is going to be located in. Is the area well lighted, but not so well lighted that glare off the CRT/TV screen will become a problem? Is the area cooled/heated to the extent that the computer (and you) will not overheat, freeze or suffer from lack of sufficient ventilation?

Are there sufficient *grounded* electrical outlets present around the room or at least in the area that will be specific to the computer work zone? What about other appliances? One common problem is to find that the outlet(s) you are on are shared with many other electrical "noise"-producing machines, such as air conditioners, heating devices and universal motor driven devices like sewing machines.

What happens when some of that "dirty" power gets into the computer system? Bad news, that's what. Lost data, periodic computer shutdowns and the like, which can be avoided if you take the time to survey the area, its power outlets and the prospects for "clean power" *before* installing the computer system.

If you have "dirty" power or a power source that is prone to voltage spikes, noise and the like, it may be to your advantage to consider the purchase of one of the low-cost power line conditioners/spike protectors on the market.

Put each piece of computer hardware on its own outlet, or, better yet, install a multiple outlet with a master power switch and don't cram or overload the outlet. Orient the computer cables and power cords of the computer system to eliminate safety hazards such as exposed connections or cords people could trip over (or

SYSTEM

By Joe Rotello

last or sometimes forgotten entirely. Keep some of the following ideas in mind when it comes time to install that new Commodore computer or change an existing installation around.

that a chair or table could pinch).

What Are We Going to Set It On?

Next, consider the tables or computer work stations you are going to use. Is each table and video monitor platform at a proper and comfortable working height? Is the table area large enough for the terminal, video monitor, printer, disk drive and other accessories that you will be using? If the printer will be placed on its own table, will the table be easy to reach when the time comes to connect cables and change or remove paper?

Be sure that the tables are sturdy. One of the most humorous (to the person who is looking on) sights is to see a hefty printer rocking a flimsy table each time the printer performs a carriage return. You might find yourself placing small wagers on when the whole installation will come grandly crashing to the floor.

So be sure to get firm, well assembled furniture. It does not have to be expensive, just well connected together and sturdy once assembled. *Don't ever* overload a table or desk *regardless* of how well it may seem to be constructed! One last point. Always plan for expansion. Someday, and a lot sooner than you think, you are going to add to the present system. Don't run out of room today for something that you will do tomorrow.

What Type of Floor?

The computer is remarkable! Remarkably sensitive to static, that is! The type of flooring that will be used in the area is very important to the life of your system (not to

Keep *all* magnetic media away from any magnetic field. That means do not store any disk or tape near loudspeakers, motors, electric typewriters, fans or other office or home equipment that generates magnetic fields when in use.

mention your sanity).

If carpeting must be used, try to select one made of low- or non-static material. Another method is to install a large area static mat over the carpet that will be centered under what one might call the "computer zone". The plastic static mat is grounded with a small slender wire and is perhaps the best safeguard against static.

An interesting option, and a good one if static will be a problem, is to locate one or more "static buttons" on the computer furniture. These "buttons", or conductive foam pads, are grounded and act to drain off any static charge that you the operator may carry. Most importantly, when you use static buttons, make it a habit to discharge yourself *before* touching a piece of equipment.

Although it seems all shoes carry some type of charge, try to *not* wear hard rubber or hard, smooth, man-made-sole shoes. They seem to be the worst as far as static is concerned. On the other hand, "Hush Puppy"™ or other similar foam or soft-sole type shoes seem to be the most compatible, static-wise, around computers.

Where Do We Store Things?

Next, take a good look at how you will store the day-to-day files, printouts and new or active-file floppy disks. A clean, dust-proof environment is essential not only to the proper operation of the computer equipment, but to the safety of your paper and critical magnetic disk media as well.

Properly covered disks should be able to stand up on shelves or in drawers. Paper should be covered and stacked vertically when possible so as to not curl or bend. Remember that both floppy disks and paper prefer dustfree environments with low humidity and moderate temperatures.

Ribbons, tapes and other accessories should be available without having to tear apart a stocking area just to find them. To keep them from drying out, store these items in plastic bags or other moisture-proof packages.

Keep *all* magnetic media away from any magnetic field. That means do not store any disk or tape near loudspeakers, motors, electric typewriters, fans or other office or home equipment that generates magnetic fields when in use. Most importantly, the telephone is a remarkable magnetic field generator! Do *not* stack or store disks near, under or directly above a telephone! When the phone rings, there goes some data.

Reminds us of the sad story we heard of the well known company that stored disks on the sides of metal cabinets using a magnetically held box...

Speaking of Telephones...

With the ever increasing use of telephone modems, it pays to take a close look at your telephone and how it relates or will relate to the

computer system. Locate the telephone in a convenient spot, not too close to the computer system, (see above) but within easy reach.

When using a modem, try to have a private, non-extension phone line. Nothing disrupts a computer/modem system more than having a person picking up an extension phone and thereby terminating your computer data transmission.

If you are in business, be sure that the modem telephone line is *not* part of your switchboard or PBX type system or your computer modem transmission could suffer the same fate as that of an extension phone. It will pay major dividends to have that extra private phone line to the outside world.

Last but not least, make sure that the modem and telephone connectors are compatible and that your local telephone company is aware that a modem will be in use. In many locales, since the telephone company central switching is done through computers, the phone company may have to "condition" their central switching unit to accept your modem data properly, or install an adaptor at or near your present phone installation.

A side note concerning modems worth mentioning is that the newer, higher speed modems can actually work with "normal" or standard phone lines such as those feeding most homes and business establishments, even though the modem data speeds have been increased. In the past, high speed modems operated properly only if the phone lines were "conditioned" or rented expressly for modem use from the telephone companies. That is generally no longer the case. The technical quality of most phone lines now exceeds that of even three or four years ago.

Last Will and Testament

The most important item is frequently the most distant in our minds. Always have complete and accurate records of model, serial number, date of purchase and replacement cost available for each and every piece of computer hardware, software and accessories you own in case the unforeseen disaster occurs. Make *many* copies of this data, keep it up-to-date and store a copy at the computer site, your home and perhaps at your bank.

Do not store this data only in a computer file! That is a dangerous mistake that we may make, thinking that if the computer is going to be used for data storage why not... well, you get the picture!

In conclusion, installing a microcomputer system, be it for use at home or in a small business or big corporation, may take more work and thought than you originally anticipated. But in the long run, you will be glad you took that extra hour or spent that extra dollar! **C**

Joe Rotello is a regular contributor to Commodore *magazine.*

In the old days (before micros, that is) you were considered "computer literate" if you knew how to give commands to a "dumb" terminal that was hooked up to the office mainframe. But in order to be "computer literate" by 1984's standards you need to know, among other things, how to choose the components of your system intelligently.

Choosing A System:

An Important Part of Computer Literacy

By Jim Gracely
Technical Editor

Once upon a time, (and still in many offices) there was a computer terminal. It had a keyboard and it had a green phosphorus or black-and-white screen. There were wires coming out of the back of it that went into the wall or into the ceiling or perhaps just ran across the floor. You logged onto it in the morning and logged off before you went home. This was, and still is, many peoples' only contact with computers.

This terminal, or "dumb terminal" as computerists refer to it, is not actually a computer at all. It is simply an appendage or "human interface". Where then is the actual computer? Ask anyone who works with a terminal of this sort, and they will refer you to the "authorized personnel only" computer room. It is inside this room that the real computer lives.

The computer room is an interesting part of any computerized office. Although only authorized people are allowed to actually enter this room, it usually has big glass windows so everyone else can look in. It is always a very clean room; it uses an excellent air filtering system. It is also a chilly room; the temperature is kept below 68 degrees Fahrenheit.

Of course, all the usual peripherals are connected to this computer. However, the disk drive and/or tape drives are about the size of washing machines, and you never have to touch them, or change them or even plug them in. They are completely serviced by the computer room personnel and they are automatically connected when you turn on your terminal.

Printers are also attached to this mainframe computer. They are also usually in the computer room and if you need a printout of something, you tell your terminal and a day later the printout mysteriously appears in a bin or is even brought to you by a low level computer room worker.

Why am I telling you all this? To show you that the definition of computer literacy is changing daily. In this office scenario, in order to be computer literate you must be able to turn on your terminal, log on and effectively access those services you need. In the days of personal computers, however, computer literacy takes on a new meaning.

First of all, what used to be your terminal is now either a Commodore 64 or VIC 20 and your television set or monitor, or a PET or CBM computer. Your peripherals all sit within six feet of your computer and include a datassette or disk drive, and possibly a printer (all of which are much smaller than washing machines). And of course the computer room is now your living room, or bedroom or den.

To be computer literate with this kind of system means understanding not only the computer, but how to load programs into the computer and run them. It also means learning how to use cassettes or disks as storage devices and how to get printouts of important documents and listings. No longer is there a computer room staff to handle all the details for you. Suddenly many people who used to consider themselves computer literate find themselves back at first grade, learning all the things they never used to need to know.

Let's take a brief look at each of these components of a computer system and find out what there is to know. I'm not going to try to teach you everything you need to know; rather I will try to point you toward the correct questions to ask.

The Computer

The computer used to be the most important part of any computer system. I don't know if that is still completely true. Depending on the application you are considering, you may be wise to buy a less expensive computer and use the savings to buy an applicable peripheral. For example, if you

want to maintain a huge mailing list but that is all you want to use your computer for, a VIC 20 would supply all the raw computing power you would need. At the same time, however, you might consider buying an IEEE-488 interface and using a more expensive CBM 4040 as a disk drive. On the other hand, if you want a smaller mailing list but intend to use it for professional-type mailings, you might use the inexpensive 1541 disk drive with a high-quality, more expensive DPS1101 printer or an IEEE interface and a letter-quality CBM 6400 printer.

The languages that are available are another important question if you are interested in learning programming. Many computers today (but not all!) are sold with BASIC built in. Want to learn COBOL or Pascal? Better do some shopping before you decide on your "perfect computer".

What you want to know, then, is: the capacity of the computer itself, what languages are built in or supported by the computer (if you want to program in APL, you should stick to the SuperPET), what peripherals are available and what interfaces are needed and available for using various peripherals. These are some of the questions you'll need answers to.

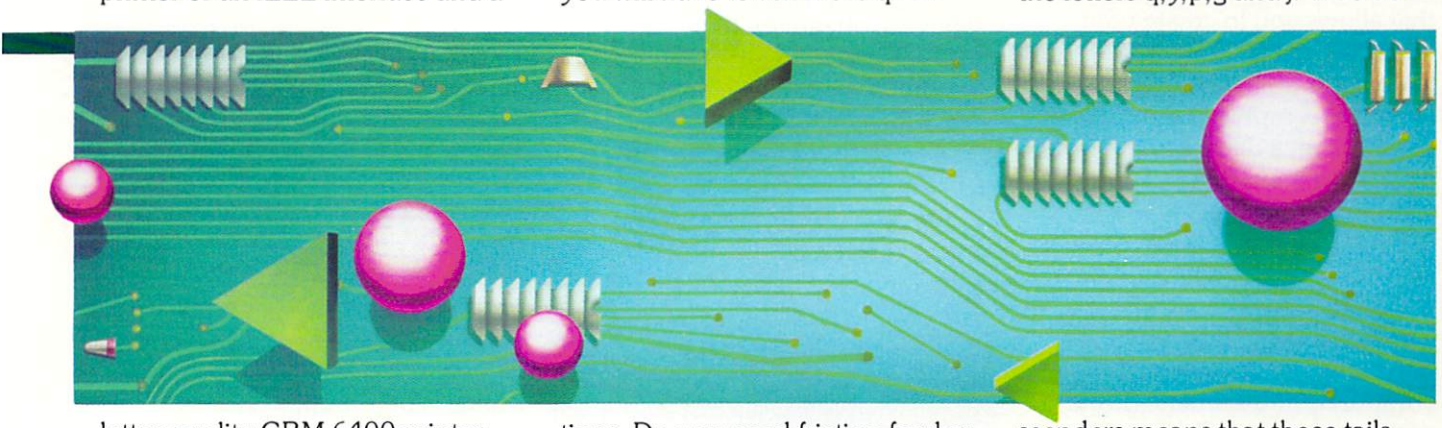
The Printer

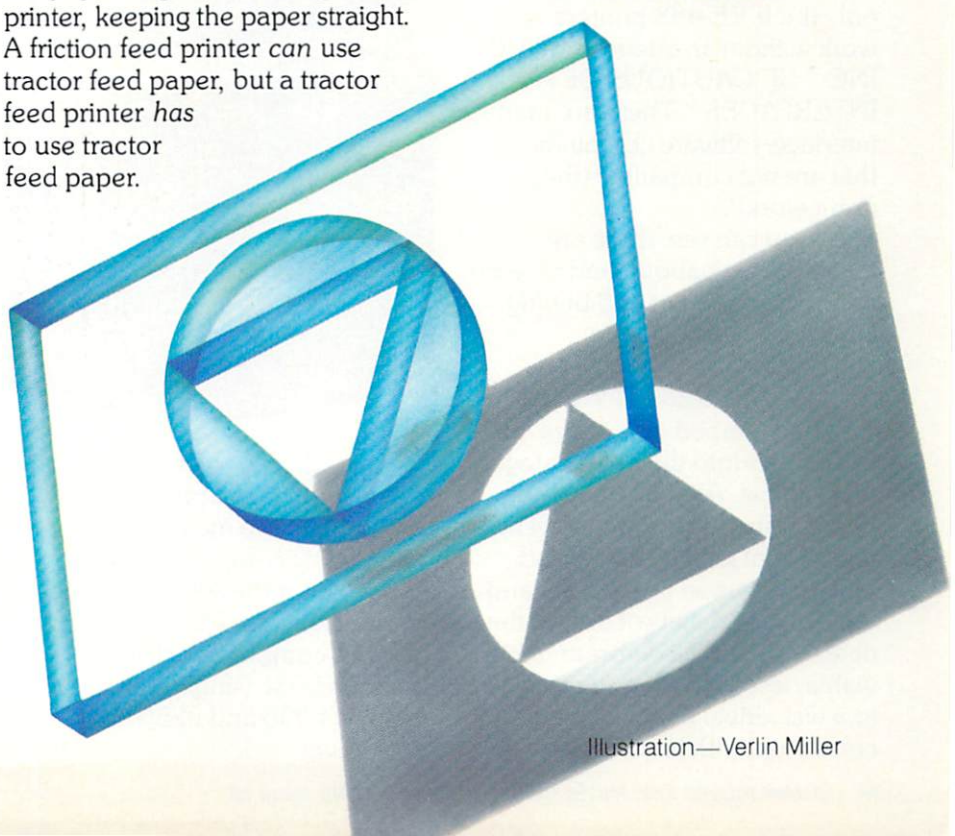
Just as the computer seems to be bowing out as the most important component of a system, the printer seems to be taking the stage. The features and abilities of printers are increasing and changing almost daily. Right now, printers include everything from dot matrix with 4" paper through eight-color dot matrix, to daisy wheel typewriter/printer combinations. Which printer is the one you need will again depend on both your budget and application.

The most obvious question in deciding on a printer is, "What are you going to be printing?" If all you want a printer for is to make personal program listings, buy the cheapest one you can find. Why spend more? On the other hand, if you want to use the printer for important business correspondence you will have to ask more questions. Do you need friction feed or tractor feed? Friction feed is like a typewriter feed where you simply roll your paper into the printer and adjust it so that it is straight. Tractor feed uses holes on the edges of the paper to guide it through the printer, keeping the paper straight. A friction feed printer *can* use tractor feed paper, but a tractor feed printer *has* to use tractor feed paper.

What about print quality? Will you be satisfied with letters made up of little dots (called dot matrix) or do you insist on a daisy wheel letter-quality printer (solid letters like a typewriter)? Going one step further, if you are satisfied with the dot matrix, how many dots would you like? The printers that use 35 dots/character are cheaper than those that use 64 dots/character. Last of all, do you like true descenders? (No that doesn't have anything to do with your family tree). This refers to the little tails on the letters q,y,p,g and j. True descenders means that those tails actually go below the bottom line of all the other letters (like they do in this magazine). Here is what the two types look like side by side:

```
quippy jaguar
QuiPPy JaGuar
```





Illustration—Verlin Miller

Unfortunately there is still a little more to think about. Paper size is another consideration. The width of paper for printers varies from 4-inch adding machine paper on a roll to 17-inch computer form paper. Some printers are adjustable and some aren't. Graphics are one more consideration when you have a printer. There are printers that are called printer/plotters (like the 1520), which have pens in them that can either draw letters for printing or draw designs. Dot matrix printers often have a custom-character feature that allows you to create your own characters. This is the feature that allows printer dumps of high-resolution graphic screens. Keep in mind that a daisy wheel printer cannot print anything other than what is on its wheel.

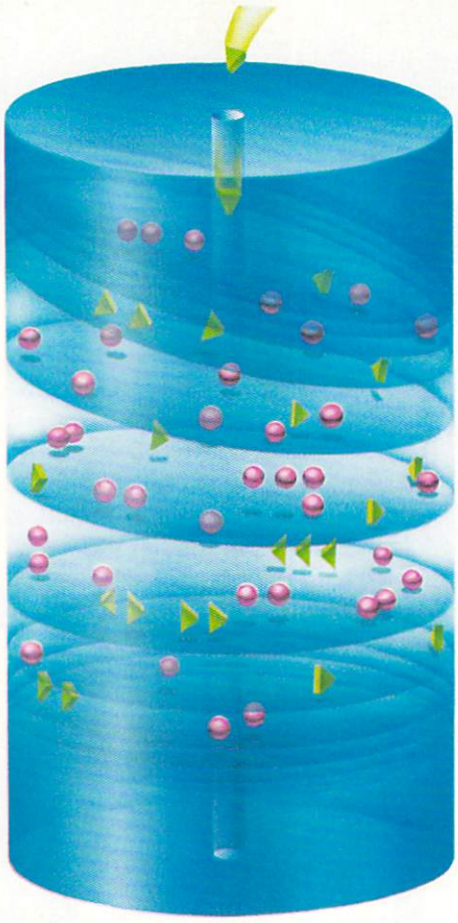
Now, last, but certainly not least, is the question of which printers will work with your computer. There are many different standards for the connection of computers and printers. Some of the types you will encounter are Commodore serial, RS-232, IEEE-488 and parallel. If you have a VIC 20 or 64, only the Commodore serial printers will work without an interface. If you have a PET/CBM or B series computer, only the IEEE-488 printers will work without an interface. WARNING * BE CAUTIOUS OF ALL INTERFACES * There are many interface/software combinations that are not compatible (they don't work!).

As you can see, there are a lot of things to think about, and to learn, before running out and buying a printer.

Storage Devices

I have lumped disk drives and datassettes into the same category on purpose. After all, they both perform the same function. The question of which is preferred again rests upon your wallet and application. Believe it or not, the datassette is not a dinosaur. It is a viable, less expensive alternative to a disk drive. If you are strictly concerned with learning BASIC and just writing some simple programs of your own, there is no reason you cannot use a datassette. I will not argue with those who explain that a disk drive is faster, has a greater capacity and is easier to use (in some instances), because it is. However, there is a level of computer usage that must be exceeded before a disk drive becomes *necessary*.

I want to mention (to avoid an avalanche of letters) that if you decide to buy a datassette and then decide you want to buy a wonderful word processor you may have a problem. Many of the better software products that are on the market are not available for use with a datassette.



If you decide to buy a disk drive, there is not presently a lot of choice in the matter. The only completely compatible disk drive available for the VIC 20 and 64 is the 1541. The PET/CBM and B series computers will work with the 2031 (single disk drive) and the 8050 and 8250 dual disk drives.

Software

Software is the newest addition to the course studies on computer literacy. New software is being introduced continuously over a wide spectrum of applications. The days when everyone saved their dollars until Atari announced its newest video game are over. Software available today ranges from expensive inferior garbage through inexpensive superior programs. Everything from games to databases are available for every computer, in every price range and again for various applications. There are database programs specifically for keeping track of your checking account, your car expenses or your cookbook. There are even programs available that will write a database program for you!

If you are about to spend some hard-earned money on a piece of software, treat it like a new camera or stereo, not like a new pair of socks. When shopping for software, the best way to find out what is good—and right for you—is to read reviews. There are software reviews in every computer publication; even some Sunday papers have software sections now.

Software is being marketed much like cameras or stereos. There are expensive places to buy these products and inexpensive places (I should add that many of the places that are a little more expensive offset this cost by offering more intelligent sales personnel and more personalized treatment). There are even sales and special offers. Keep your eyes open, find out what product has those features you are most interested in and then start shopping for it.

That's about all I'm going to say about what computer literacy is. As you can see, there is much more to being computer literate than being able to write a program to print your name on the screen ten times. In fact, becoming computer literate is much like going to school; depending on what you want to do, you can graduate at a number of different levels. C

Computer Limitations and Capacities

Many people share one of two popular opinions about computers. The first is that computers are nice. They are good for playing games or programming on, but they aren't very useful for everyday applications. The second opinion is that computers are miracle machines. Everything you do can be accomplished hundreds of times faster and easier with a computer. In reality, computers lie somewhere between these two extremes.

The personal computers of today are number-handling machines. What is the significance of this? Well, it means that anything you tell the computer to do is broken down into a series of numbers. Whether you are computing the cube root or 324.4 or what "name no one man" spells backwards, the computer does everything with numbers. The result of this is that working with numbers is usually a little easier for the computer than working with letters and words. Perhaps sometime in the future there will be computers that are word handlers, but for now we'll have to settle for numbers.

The time it takes the computer to perform any small calculation is very fast. Many people think that it does simple calculations instantly. They then reason that a big calculation can be broken down into a bunch of simple calculations. The last logic step is to figure that a bunch of instants is still an instant. Now they have convinced themselves that the computer can do anything in an instant.

Alas, this reasoning is false. The time for a simple calculation is indeed very fast, but not instantaneous. For example, it takes the computer about one second to add all the numbers from one to 300. If we add all the numbers from one to 5000, we use up over 18 seconds. This is hardly an instant! When working with a program that is performing some large task (such as sorting 500 addresses on a disk), be patient. There are limits even to the computer.

Memory

Many people wonder about the mysterious "byte". By now, most people know that its definitions would include: eight bits of binary code, or the "word" of the computer. But the next question that arises is, "What is it for?" or what does one byte represent to the computer user?

One byte can represent any of a number of things.

In the simplest sense it represents an ASCII character. This is a set of 128 letters, numbers and symbols (+,−,<,>...) that is defined for all computers. Commodore ASCII adds an additional 128 characters for the graphic symbols and reverse characters. If you have 600 bytes of memory, you can store a piece of text or a letter that contains 600 characters (including spaces and carriage returns). The video screen is also part of memory and therefore each character on the screen also uses one byte of memory.

A second use for a byte is for representing a BASIC "token". Each command in BASIC is stored in a Commodore computer (this isn't standard for all computers) as a single byte. The BASIC command PRINT, for instance, is stored as the number (or byte) 153. This conserves memory and allows you to fit a longer BASIC program into a given number of bytes than would otherwise be possible.

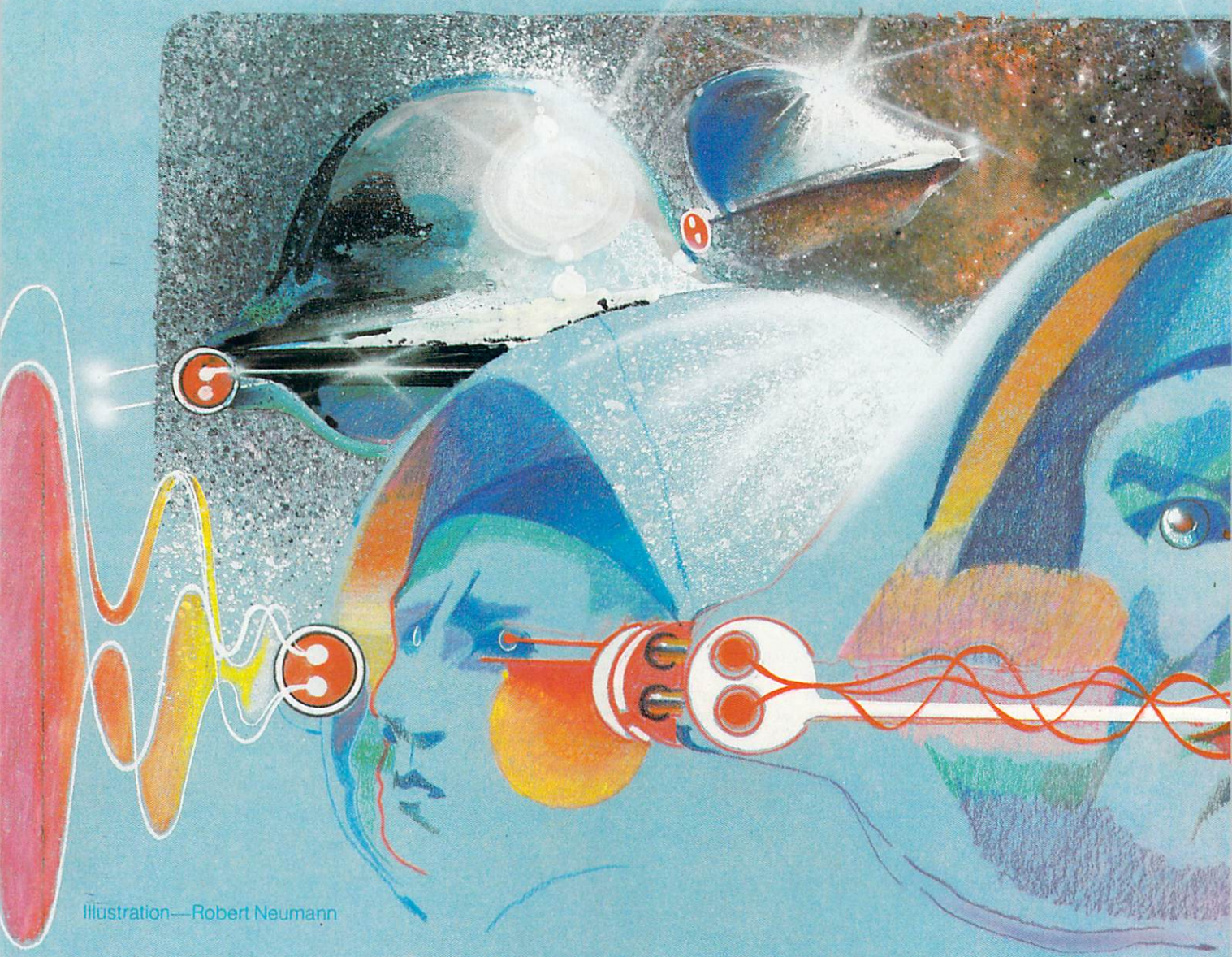
Another use for a byte is to represent one empty memory space. This space may be filled with an ASCII character, a BASIC token or one byte of a machine language program. This is the most common use of the term when referring to storage devices. For example, a 1541 diskette has a capacity of about 170,000 bytes. You can fill this space with whatever you wish.

The next magnitude up from a byte is a kilobyte. This is 1024 bytes (it is not exactly 1000 bytes as many people believe). The number 1024 comes from two raised to the 10 power (2*2*2*2*2*2*2*2*2*2). This is simply an easier way to talk about a large quantity of bytes, just as dollars are easier to deal with than pennies. A capital K is the symbol for a kilobyte and this is the way it is usually represented. If a computer has 1024 bytes of memory, it is referred to as a 1K computer.

The 64 has 64K (or 65536 bytes) of memory that can be used, hence the name. I will admit that this is a lot of memory, but compared to what? Compared to the VIC 20, this amount of memory will hold an enormous amount of data. Compared to a good dictionary this is very little memory. Consider a pocket dictionary with 100,000 entries. Each entry probably has about 100 characters in its definition. This means the dictionary contains over 10 million characters! This is about 10,000 bytes. It would take the memory of more than 152 64's to hold all of these characters. Even if it were on diskettes, with 170K per disk, you would need almost 59 disks to hold it all.

As you can see, although computers are becoming faster with more memory, there are still limits to their capacities. People keep wondering how long it will be before the computer passes the human mind in capacity. Well, most sources I've seen put the capacity of the brain at about 10 billion neurons (just think bits). This is still quite a magnitude above anything available today. As for tomorrow... **C**

—Jim Gracely

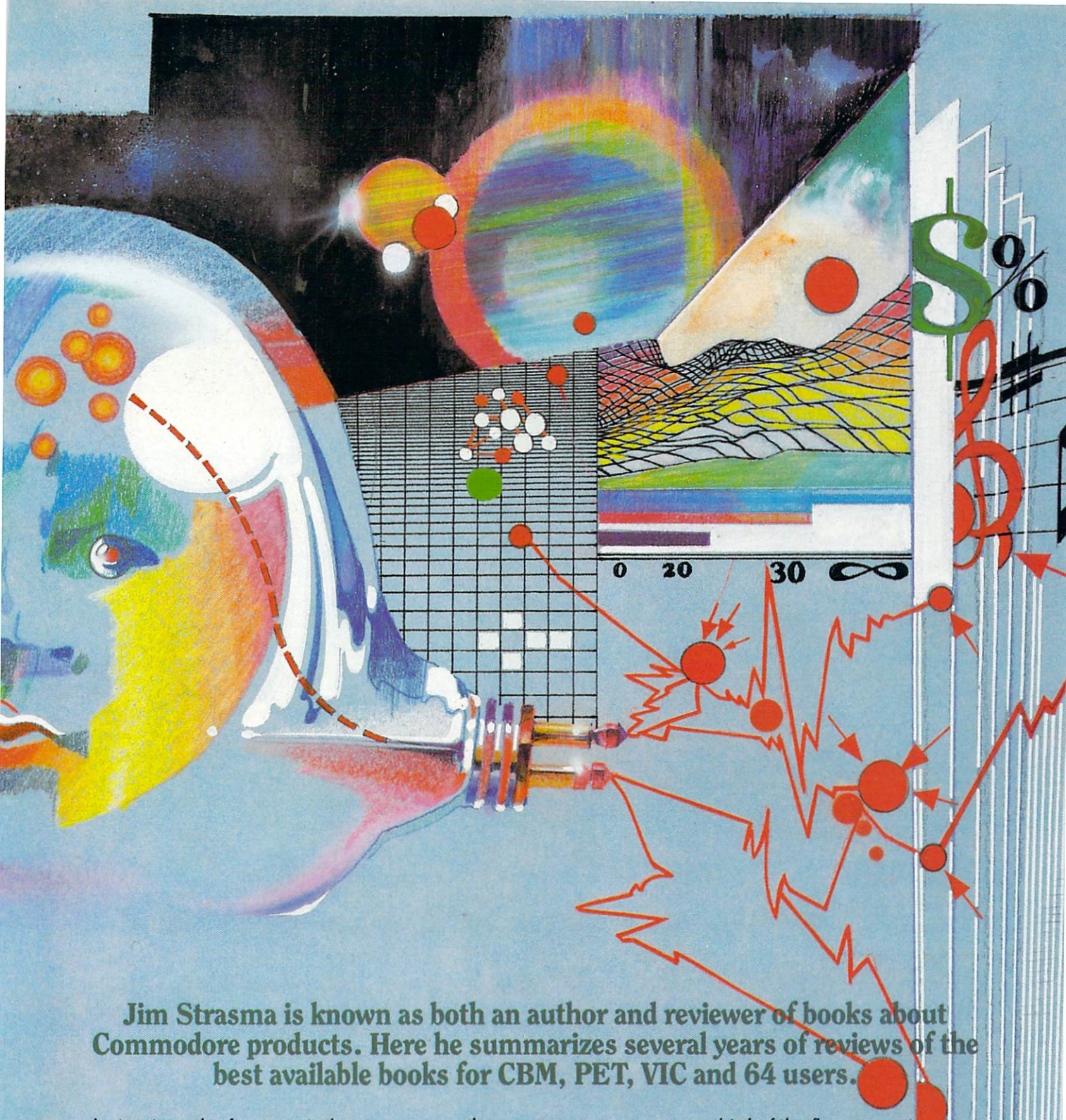


Illustration—Robert Neumann

books

Still a Main Road to Computer Literacy

By Jim Strasma



Jim Strasma is known as both an author and reviewer of books about Commodore products. Here he summarizes several years of reviews of the best available books for CBM, PET, VIC and 64 users.

An ironic truth of our age is that the technology widely expected to doom the book industry instead brought it roaring out of a deep recession. Computers, which theoretically eliminate the need for paper reference materials, have created a market for an incredible variety of books. According to various surveys, the average computer buyer now buys an average of ten computer books, in hopes of becoming "computer literate". In response to the rapid increases in both computer and computer book sales in each of the past eight years, nearly every publisher now offers computer titles. Up to one-third of the floor space in some bookstores is devoted to computer books and customers are still buying nearly all of them.

Unfortunately, many of the newly-offered computer books are of little or no value. "Instant experts" who have owned a microcomputer for a few months at

most, spout truths that aren't so and publishers who know nothing about the technology they cover are unable to even detect, let alone correct, the errors. Worse yet, some offerings don't even attempt to achieve helpfulness, merely sales.

Many expect 1984 to be the year in which computer books suffer the shake-out already underway for the computers they describe. Though I'm responsible for some of the glut (as will be obvious below,) I'm hoping we'll soon see fewer and better resources for computer literacy.

Meanwhile, here are brief reviews of several of the best books currently available for Commodore computers, whether PET, CBM, VIC or 64. They are adapted from the pages of our *Midnite Software Gazette*, a bimonthly review magazine that has covered Commodore products since 1978. Unless otherwise labeled, the reviews are largely mine, though most include important thoughts from readers. For reviews of new books as they appear, you may wish to subscribe. Sample issues are available on request from: Midnite, 635 Maple, Mount Zion, IL 62549.

Now, without further ado, our list of favorites:

Before BASIC

User's Guide to Commodore 64 and VIC 20 Computers, Software & Peripherals, by Jim & Ellen Strasma for Consumer Guide. For VIC and 64. Teaches computer use, not programming.

This is a book Commodore could have packed with the VIC and 64. Excellent information and illustrations, from unpacking your computer and putting it all together to short summaries of useful programs. If I were a Commodore dealer I'd buy a pile of them, give 'em away free with the computer and save myself a lot of questions. Highly recommended.

—Brent Anderson

Easy BASIC

Armchair BASIC: An Absolute Beginner's Guide to Programming in BASIC, by Annie and David Fox. From Osborne/McGraw-Hill.

Simple enough to use in elementary schools. Also great for people who know they should learn to program their computer but are afraid. Its only threatening feature is a short quiz after each chapter, but the answers are in the back. The book's prime weakness is its failure to show structure in programs. It also omits files and nearly all functions. Although not specifically for Commodore, its BASIC is limited enough to work well on Commodore machines. Definitely worth considering.

Hands-On BASIC With a PET, by Herbert Peckham. From McGraw-Hill.

A helpful beginner's workbook for students who learn best by working their way through a series of problems. Avoids heavy-duty math, and includes enough humor to maintain interest. Wire-bound to lie flat by your computer. Primarily for older PET models.

Kids and the VIC and **Kids and the 64,** by Edward H. Carlson. From Datamost. Wirebound.

Aimed at the seventh-grade reader with a datassette, it succeeds admirably with adults as well. My college classes are captivated by its simple, but not simplistic style and imaginative illustrations. Each chapter includes instructor notes, questions and assignments (with answers in the back.) Short chapters cover most important keywords, but not rarely-used functions or file commands. There is also a glossary, list of error messages and special symbols in the back. Highly recommended.

My Computer Likes Me When I Speak in BASIC, by Bob Albrecht. From Dilithium Press.

The book that taught me BASIC. What more can I say? OK, so it's 12 years old and only 61 pages long—it's still one of my favorites, managing to cover everything from getting started to two-dimensional arrays, subroutines and loops, aided by the plentiful graphics for which Albrecht is known. Recommended.

PET BASIC: Training Your PET Computer, by R. Zamora, B. Albrecht and W. Scarvie. From Reston Publishing.

An excellent manual with plentiful and well-done illustrations. However, it has three major faults. First, the small calculator keyboard, which has been unavailable for years, is used throughout. Second, only at the book's end do you learn VERY briefly how to save a program or load it from cassette, without a word on reading or writing cassette data. Third, nowhere are disks or printers mentioned. Primarily for beginners.

Your Commodore 64 and **The VIC 20 User Guide,** by John Heilborn and Ran Talbott. From Osborne/McGraw-Hill.

These are stripped down editions of Osborne's popular guides for the PET. Simpler to read than the earlier three editions but less comprehensive. Their best feature is a clear discussion of graphics.

Full BASIC

CBM Professional Computer Guide and **PET Personal Computer Guide,** by Adam Osborne & Jim & Ellen Strasma. From Osborne/McGraw-Hill. For 80- and 40-column models respectively.

This book is strongly recommended to CBM owners who want to know more about their computers than is contained in the Commodore manuals. The depth of material is impressive and shows a long and close connection with the Commodore machines. If you're a CBM owner, keep this book next to your computer because you'll turn to it often.

—Personal Computing 11/83

One of the most well-written and organized computer manuals I have seen. It would be impossible to use the CBM computer effectively without it. *—David C. Ayers*

Elementary BASIC, by H. Ledgard & A. Singer. From SRA.

You'll stay up all night to read it. Though not specifically for PET, this is an excellent intermediate text on BASIC. Written in a story form using Sherlock Holmes and his mysteries, the book makes great reading. It also teaches, painlessly, only the best techniques of structured programming.

Elementary Commodore 64, by William B. Sanders. From Datamost.

Light and conversational. Good use of line drawings but very poorly printed. Helpful demo programs scattered throughout. Numerous errors in early printings and weak on files, but very popular.

Instant Freeze-Dried Computer Programming in BASIC, by Jerald R. Brown.

A clear and compact learning and reference guide. Printed in black and red, with arrows, cartoons and graphics, all of which make it easy to see what is and isn't important.

Introduction to BASIC, Part 1 and 2, by Andrew Colin. From Commodore. Versions for VIC and 64.

From England. Excellent textbook for self-study. Full coverage of BASIC along with proper structure (even includes flowchart template.) Consists of a well-illustrated wirebound workbook and two tapes of utility programs and quizzes. Recommended.

Advanced BASIC

Advanced BASIC Programming for the Commodore 64, by Michael Richter. From Brady.

Good short book for the serious programmer, with practical suggestions for organization and design in clear, precise English. Many tips and techniques. Helps avoid pitfalls in writing data bases, inventory programs, games, etc. Includes useful programs. To be read more than once.

—Dr. Elizabeth Kaspar

Programming the PET/CBM, by Raeto West. From Level Limited or COMPUTE! Books. BASIC and assembly-language PET guide.

Expert-level info on programming in BASIC and 6502 machine-language for PET/CBM computers. *Not* for beginners, but leaves virtually no high-level programming stone unturned. Anyone deeply enough into PET to benefit from this book will use it daily. Anyone who programs CBM or PET computers for a living MUST get a copy, preferably of the far better-bound Level Limited edition.

Mastering the VIC 20, by Jones, Coley and Cole. From Wiley. Diskette of 34 useful programs sold separately.

For readers ready for PEEK and POKE. The eight chapters cover the gamut from simple VIC BASIC to speed techniques, special effects and hexadecimal; memory configurations, graphics, ports, peripherals, architecture and machine code. Considered by many to be the best book out for VIC.

Basic Reference

BASIC Handbook, Second Edition, by David Lien. From Compusoft.

For anyone who often converts programs from one dialect of BASIC to another. Covers some 250 different keywords, and the details of their use in nearly all popular microcomputers. Better yet, includes substitutes to use when your BASIC lacks a needed keyword. Overpriced for casual users, but should be on every pro's shelf.

Programmer's Reference Guide. From Commodore and Howard W. Sams. Versions for VIC and 64.

Good reference books for skilled users. Covers programming, machine language, graphics, and I/O, plus a hardware schematic. A very informative book, although it has a few confusing sections. Recommended.

Machine Language

6502 Assembly Language Programming, by Lance Leventhal. From Osborne/McGraw-Hill. 606 pages.

The best book for machine language programmers to use in learning 6502 assembly language, and an excellent advanced text for others. Each instruction is covered in detail, with many, many examples (all tested on an Apple.) Also covers I/O chips. Recommended.

6502 Micro Chart, from Micro Logic Corp.

This instruction set summary is the *best.* A hard plastic notebook sheet with the most information in one place you will ever find on the 6502 and everything you will be likely to need. Highly recommended.

Machine Language for Beginners, by Richard Mansfield. From COMPUTE! Books. For all models.

An understandable short course in 6502 machine language, presented through analogy. Abounds in illustrations, standards and examples. Useful appendix is larger than the text. Written in an easy-to-handle, durable, graphically pleasant, spiral-bound format that unfortunately tries to cover several brands at once, leading to mild confusion at times. Avoid its BASIC mini-assembler and other programs in favor of Micromon (included).

Other Languages

COMAL Handbook, by Len Lindsay. From Reston Publishing.

Most complete description of COMAL anywhere. Every keyword is listed alphabetically on a separate page, defined at length, and shown in at least one *useful* program, with extensive references to related words and other examples. Includes *all* BASIC keywords, with references to appropriate COMAL substitutes. Appendices contain a complete guide to COMAL structures grouped to show their relationships.

Osborne CP/M User Guide, by Thom Hogan. From Osborne/McGraw-Hill.

Understandable, even for beginners, and quite complete in explaining options. Skilled users may safely drop in just at the page containing the command needed. Well bound and printed for a paperback. If you use one of the CP/M attachments for the PET or 64, you need it.

Reviews

Best VIC/64 Commodore Software, by Jim and Ellen Strasma. From Consumer Guide.

Lists and reviews the top software programs available for the 64 and VIC. Includes composite user group ratings. Just out and popular.

Commodore Software Encyclopedia, Third Edition. From Commodore.

Whole PET Catalog, by the Strasmas with Bruce Beach. From Midnite Software. Hundreds of reviews and hints for users of CBM, PET and VIC computers.

I hope to be among the first to congratulate you fine folks on the excellent job you did in putting *The Whole PET Catalog* together. There are no other words to say, except you just did a tremendous job and I hope it was a labor of love. —*Slim Wilson*

Hardware

PET and the IEEE Bus, Second Edition, by Fisher and Jensen. From Osborne/McGraw-Hill.

Great for persons wishing to know how PET communicates with the IEEE-488 bus. Has IEEE to RS-232 interface circuit and programming examples. All bus transactions are explained in detail. —*Arthur Cochrane*

Reference list of commercial products and vendors for all models. Product descriptions appear to be advertisements, not reviews. No visible effort to weed out duds, although those products labeled "Commodore Approved" and Commodore's own products are usually preferable to others.

PET Interfacing, by Downey and Rodgers. Blacksburg Series from Howard Sams. Primarily for PETs with upgrade BASIC 2.

An understandable guide to the user port, memory expansion connector and IEEE-488 bus. Includes several hardware projects (of varying quality.) Excellent discussions of the user port, address decoding, many common 7400 series TTL integrated circuits, and IEEE handshaking. Its programs include source code. Requires some knowledge of wire wrapping and 7400 series integrated circuits.

VIC 20 Connection, by James W. Coffron. Excellent guide for the intermediate hacker. The 273 pages describe methods and include schematics for "real world" projects.

Begins at a very basic level, describing bit-level Input/Output, continues with binary weights and programming. Ample listings and examples. Those familiar with breadboarding will have no problem. Develops a home control and security system that includes a voice synthesizer, and covers interfacing to slow devices.

—Hank Mroczkowski, Commodore Houston User Group

Programs

Sprite Graphics for the Commodore 64, by Sally Greenwood Larsen. From Prentice-Hall/MicroText.

Good explanation of how to create sprites. Fairly smooth-running sprite graphics sampler program that needs some slight correction. Extremely helpful for the beginning graphics designer.

—Michael Voight

Library of PET Subroutines, by Nick Hampshire. From Hayden Publishing. For PET Upgrade BASIC 2.

Dozens of pre-written useful BASIC and assembly language subroutines, with full documentation. Includes routines for data entry, sorting, plotting and other topics. Program disk also available.

This is one of a series of Nick Hampshire publications for the PET, VIC and 64, all published by Hayden. The series covers useful topics, and is available as a set from Commodore, in corrected editions.

PET Fun and Games and **64 Fun and Games,** by Ron Jeffries and Glen Fisher. From Osborne/McGraw-Hill. Disk copy of the games available directly from The Code Works in Goleta, California.

Thirty-one favorite games from *Cursor* magazine in versions that work on both 40- and 80-column models. Has accurate, easy-to-read listings and is well illustrated, plus has Osborne's usual high quality printing and binding. Recommended.

PET Games and Recreations, by Lindsay, Oglesby and Kunkin. From Reston Publishing. Cassette of longer programs available.

Twenty games for graphic keyboard PETs. Most are worth keying in and provide enjoyment for all ages, but none have sound!

Excellent BASIC structure and program explanations. The humorous lectures at the end hide useful programming hints compatible with all PETs.

Other

Computer Phone Book, by Mike Cane. From Plume.

A unique and excellent reference on telecommunicating. Starts out with equipment and programs needed, including simple explanations of key terms. Then describes available on-line systems with tips. Standard control codes are thoroughly explained. Also shows example screens.

Heart of the book is a comprehensive and extensively cross-referenced directory of more than 400 systems and services. Optional monthly updates are promised.

—Paul Motise

COMPUTE!'s (X-th) Book of (you name it). From COMPUTE! Books.

These popular books, in PET, VIC and 64 versions, reorganize and reprint the best articles from back issues of COMPUTE! in a handy wire binding, though reportedly without correcting some errors in the original articles. Handy to have around.

Reader reviews of other books are welcome. Address mail to:

Rev. Jim Strasma
1238 Richland Ave.
Lincoln, IL 62656 USA

Include a stamped self-addressed envelope if you wish a reply. **C**

commodore 64

MAGIC DESK I

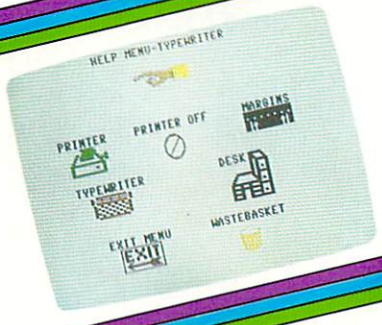
TYPE AND FILE

Only from Commodore—the excitement and simplicity of Magic Desk

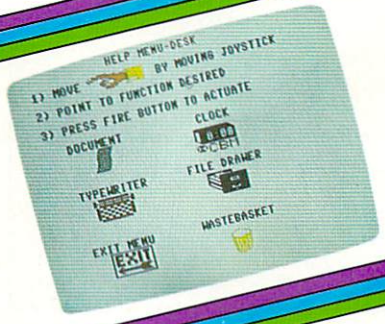
Only Commodore brings you the magic of MAGIC DESK... the next generation of "user friendly" software! Imagine using your computer to type, file and edit personal letters and papers *without learning any special commands!* All MAGIC DESK commands are PICTURES. Just move the animated hand to the picture of the feature you want to use (like the TYPEWRITER) and you're ready to go.

The MAGIC DESK Typewriter works just like a real ELECTRIC TYPEWRITER ... and it's COMPUTERIZED. All the filing is *electronic*. Excellent sound effects and screen animation make typing fun, whether you're typing letters, reports or memos ... and the built-in filing feature makes MAGIC DESK useful for keeping names and addresses, home inventory lists, insurance information and more.

Your COMMODORE 64, COMMODORE DISK DRIVE and MAGIC DESK are an unbeatable combination. Filing operations are automatically linked to your Commodore disk drive—but you don't have to know any commands—just "file" the pages you type in the file cabinet and your text is automatically saved on diskette. There are 3 file drawers with 10 file folders in each drawer and 10 pages in each folder.



To PRINT a page you've typed, just "point" at the picture of the printer and your pages are automatically printed on your COMMODORE PRINTER or PRINTER/PLOTTER. If you want to erase what you've typed, the WASTE-BASKET under the desk lets you "throw away" pages. There's even a DIGITAL CLOCK which helps you keep track of time while you're typing.



Not only is MAGIC DESK easy to use... it's hard to make a mistake! Just press the COMMODORE key and one of several "help" menus" appears to tell you exactly what to do next. Special messages show you how the various picture commands work and help you when you make a mistake. Help messages also show you how to use the printer, filing cabinet, digital clock and wastebasket.

commodore COMPUTERS

First In Quality Software

Captain COMAL

(continued from page 33)

```
  get'type
  get'name
  get'blocks
  file'info$:="### "+this'file$+" "+type$(this'type)+" ###"
  print using file'info$: file'count+1,blocks
  if file'count>=max'files then
    print "too many files"
  elif this'type<>128 then
    file'count:+1
    blocks'free:-blocks
    file'name$(file'count):=this'file$
  endif
  check'dir'cycle
endproc next'file
//
proc get'type
  this'type:=disk'get(master'file,file'end)
  disk'get'skip(2,master'file,file'end)
  if this'type<129 or this'type>132 then this'type:=128 //fix type
endproc get'type
//
proc get'name
  disk'get'string(this'file$,16,master'file,file'end)
  // next 3 lines convert shift spaces to normal space
  for space#:=1 to len(this'file$) do
    if ord(this'file$(space#))=160 then this'file$(space#):=chr$(32)
  endfor space#
  disk'get'skip(9,master'file,file'end)
endproc get'name
//
proc get'blocks
  blocks:=disk'get(master'file,file'end)
  blocks:+256*disk'get(master'file,file'end)
endproc get'blocks
//
proc check'dir'cycle
  block'flag:=true
  block'count:+1
  if block'count=8 then
    block'count:=0
    block'flag:=false
  endif
  if block'flag then disk'get'skip(2,master'file,file'end)
endproc check'dir'cycle
```

```
//
proc sort'dir
  input "Sort directory?": text$
  if text$="Y" or text$="y" then
    quicksort(file'name$,1,file'count,16)
  endif
endproc sort'dir
//
proc print'dir(ref d$(),start,num'cols,num'rows)
  start:-1 //allow for disk name as first item printed
  d$(0):=date$+" id:"+disk'id$
  for row:=0 to num'rows-1 do
    for col:=0 to num'cols-1 do
      this'one:=start+(col*num'rows)+row
      if this'one<=file'count then print tab(1+(17*col)),d$(this'one),
    endfor col
    print
  endfor row
  print  // form feed could go here
  start:+(num'cols*num'rows) //start point for next page
  if file'count>=start then
    d$(start-1):=disk'id$+" continued "+disk'id$
    print'dir(d$,start,num'cols,num'rows)
  endif
endproc print'dir
//
// 4 disk'get routines:
//
func disk'get(file'num,ref file'end) closed
  poke 2026,file'num
  sys 2025
  file'end:=peek(144)
  return peek(2024) //value of character
endfunc disk'get
//
proc disk'get'init closed
  for loc#:=2024 to 2039 do
    read v
    poke loc#,v
  endfor loc#
  data 0,162,0,32,198,255,32,207
  data 255,141,232,7,32,204,255,96
endproc disk'get'init
//
```

```
proc disk'get'skip(count,file'num,ref file'end) closed
  for x#:=1 to count do y:=disk'get(file'num,file'end)
endproc disk'get'skip
//
proc disk'get'string(ref item$,count,file'num,ref file'end) closed
  item$:=""
  for x#:=1 to count do item$(x#):=chr$(disk'get(file'num,file'end))
endproc disk'get'string
//
// 3 quicksort routines:
//
proc quicksort(ref a$(),left',right',reclen) closed
  dim pivot$ of reclen, buffer$ of reclen
  partition(left',right',left',right') // sort a$(left':right')
endproc quicksort
//
proc partition(left',right',i,j)
  pivot$:=a$((left'+right') div 2) // get middle element as pivot
  repeat  // perform swappings
    while pivot$>a$(i) do i:+1
    while pivot$<a$(j) do j:-1
    if i<=j then swap(a$(i),a$(j)); i:+1; j:-1
  until i>j
  if left'<j then partition(left',j,left',j) // sort a$(left':j)
  if i<right' then partition(i,right',i,right') // sort a$(i:right')
endproc partition
//
proc swap(ref a$,ref b$)
  buffer$:=a$; a$:=b$; b$:=buffer$
endproc swap
```

Translating Machine Language Routines Into BASIC Loaders

by M. W. Caprio

Although a great deal of the Commodore 64's power can be unleashed with the BASIC language, many programmers eventually find that certain routines that they want to include in their programs really require machine language to run properly. This does not mean that they will have to abandon BASIC entirely; a main program can still be written in BASIC and can call specific machine language subroutines to perform tasks that would be too slow, too cumbersome or impossible for BASIC to do. BASIC programs that use machine language subroutines are sound solutions to programming problems. They blend the best of both worlds and demonstrate a certain hybrid vigor.

Machine language subroutines are usually written with a programmer's aid like Commodore's *Assembler Development System,* or they may be coded with a machine language monitor program like *SuperMon* 64. The point is that they are composed apart from the BASIC program that they will ultimately serve. Once written, the machine language subroutines are saved as a separate entity. When the main (BASIC) program runs it must "boot" the machine language routine into memory before it may call for its services. Many programmers, however, would prefer that the main program include the machine language instructions in the form of a BASIC loader. It is a reasonable preference with several notable advantages.

For the uninitiated, a BASIC loader appears as a sequence of data statements in a BASIC program. It contains the decimal equivalents of the hexidecimally coded machine language instructions. The machine language program is loaded by the "loader" sequentially reading those data values and POKing them into the memory locations that the machine language routine is to occupy.

If a BASIC loader is included in the main program as the means of loading its machine language subroutines, then both the main program and its subroutines can be saved in a single operation. There will be no need to handle the machine language subroutine as a separate program any longer. Copies can now be made much more quickly. Then too, a printed program listing will contain the entire program, complete with all its subroutines. Any typist can key the listing into a computer. No part of it requires the use of a machine language monitor to get it into memory or to save it.

Manually constructing a BASIC loader can be a tedious job. You would do it by first PEEKing through a machine language program that is in memory to get the numerical data that constitutes the program's instructions. Then you would type up DATA statements that include these values. A five hundred-byte program would have to have five hundred items carried in those DATA statements.

In addition to being very time consuming, any typographical error that you may make transcribing the data can be extremely difficult to locate. The process is roughly equivalent to the ordeal of copying a BASIC loader from a magazine article. When you are getting something new, the typing effort may be worthwhile and certainly takes less time than it would to create the entire program yourself. But when you already have the program in memory and all you need is the same program but in the form of a BASIC loader, then doing all that typing becomes a tiresome redundancy.

This utility program—I call it TRANS$C000—will read through a machine language program in memory and will construct the data statements for you. After it runs, you will be left with a list of numbered data statements in RAM that may be handled as you would handle any data statements that you had typed in yourself.

The bad news is that if you want to try this, you will have to type it in. Once that is done, you must remember to save it. Don't even test it—*save it immediately.* As this program runs, it is overwritten by the data statements that it creates, so testing it without first saving a copy will give you the opportunity to do all that typing a second time.

To use TRANS$C000 you should first have a ma-

chine language program loaded that you want to translate. If you don't, though, it will work anyway, translating whatever it finds in the memory locations that you point it to. As it runs TRANS$C000 will ask for you the starting and ending addresses of the machine language program to be translated and the statement number that you would like assigned to the first data statement. (Thereafter they will be numbered sequentially with an interval of one.)

Press RETURN and the next thing you will see is READY at the top of the screen. It's done; it's fast.

Type LIST now and TRANS$C000 will be gone. In its place you will have a list of data statements. These are the DATA statements for the BASIC loader of your program. Add a FOR-NEXT loop that reads the data and POKEs it into memory, and your loader will be complete. You can begin to build the rest of your BASIC program around it now or, if you have the necessary utility, you can save it and later append it where necessary.

TRANS$C000 was originally written with a separately loaded machine language subroutine. It was then used to translate itself into the BASIC loader that is now part of the program listing that you see here. It works.

The machine language subroutine that TRANS$C000 uses occupies memory locations from $C000 to $C1EF (49152 to 49647, decimal). If you need to translate a program that overlaps that section of memory, then this version of the translator program will not work. But there is a solution.

Program 2 is called TRANS$8000 and is identical to TRANS$C000 except that its machine language subroutine is stored between $8000 and $81EF (32768 to 33263, decimal). If TRANS$C000 conflicts with the program that you want to translate, then TRANS$8000 probably won't. For maximum versatility you will want both versions of this program in your library.

The translator program includes a BASIC component and that part of it sits in BASIC's RAM. This is true of both versions, but it shouldn't create a problem because the machine language programs that you will be translating will have been written to be out of BASIC's way. This is a reasonable surmise, considering that you will have planned them to be used with BASIC loaders.

Program 1 and Program 2 are the listings of TRANS$C000 and TRANS$8000, respectively. The sample run is included to show you what to expect during a typical run. The data statements created by this run reflect the contents of addresses 49152 through 49175, the first twenty-four addresses of TRANS$C000's own machine language subroutine.

If you want to make BASIC loaders of your machine language routines, the TRANS program can be a real time saver. With it, a five hundred-byte machine language program can be translated into the data statements of its BASIC loader in about five minutes, including loading and typing times. The actual run time for such a job would be less than one second. Keying in the TRANS programs will be a good investment of your time. C

```
SAMPLE RUN

RUN

WAIT WHILE THE BASIC LOADER RUNS

PRESS RETURN TO CONTINUE

STARTING ADDRESS (DECIMAL)? 49152

ENDING ADDRESS (DECIMAL)? 49168

IS THAT RIGHT? Y

WITH WHAT NUMBER DO YOU WANT YOUR
DATA STATEMENTS TO BEGIN? 1000

READY

LIST
 1000 DATA232,3,1,8,2,8,19,8,8,
192,17, 192,12,0,0,0,0
 1001 DATA255,169,1,141,2,192,141
```

TRANS$C000

```
1000 REM ** TRANS$C000
1010 REM ** M.W. CAPRIO 1/5/84
1020 POKE 53281,6:POKE 53280,14
     :PRINT CHR$(154)
     :REM ** SET COLORS TO DEFAU
     LT VALUES
1030 PRINT"[CLEAR,SPACE4]
     WAIT WHILE THE BASIC LOADER
      RUNS"
1040 FOR I=49152 TO 49647:READ A
     :POKE I,A:NEXT
     :REM ** LOAD ML SUBROUTINE
1050 PRINT"[DOWN,SPACE8]
     PRESS RETURN TO BEGIN."
1060 GET C$:IF C$<>CHR$(13)THEN
     1060
1070 INPUT"[CLEAR]STARTING ADDRE
     SS (DECIMAL)";S
1080 INPUT"[DOWN]ENDING ADDRESS
     (DECIMAL)";E
1090 IF S<0 OR E<0 THEN PRINT"
     [DOWN]ILLEGAL QUANTITY ERRO
     R":GOSUB 1280:GOTO 1070
1100 IF S>65535 OR E>65535 THEN
     PRINT"[DOWN2]ILLEGAL QUANIT
     Y ERROR":GOSUB 1280
     :GOTO 1070
1110 IF S>=E THEN PRINT"[DOWN]
     END ADDRESS TOO LOW ERROR"
     :GOSUB 1280:GOTO 1070
1120 INPUT"[DOWN]IS THAT CORRECT
     ";C$:REM ** A     SECOND CH
     ANCE TO CORRECT ERRORS
1130 IF LEFT$(C$,1)<>"Y"THEN 1070
1140 PRINT"[DOWN]WITH WHAT NUMBE
     R DO YOU WANT YOUR"
1150 INPUT"DATA STATEMENTS TO BE
     GIN";D
1160 IF D<0 OR D>65535 THEN PRIN
     T"[DOWN]ILLEGAL QUANTITY ER
     ROR":GOSUB 1280:GOTO 1070
1170 :
1180 REM ** COMPUTE HIGH AND LOW
      HEX EQU-          VALENTS
      OF THE VARIABLES
1190 SH=INT(S/256):SL=S-SH*256
1200 E=E+1:EH=INT(E/256)
     :EL=E-EH*256
1210 DH=INT(D/256):DL=D-256*DH
1220 :
1230 REM ** PASS STARTING AND EN
     DING              ADDRESS
     ES AND DATA STATE-
1240 REM    MENT NUMBER TO ML SU
     BROUTINE
1250 POKE 49160,SL:POKE 49161,SH
     :POKE 49162,EL:POKE 49163,EH
1260 POKE 49152,DL:POKE 49153,DH
1270 SYS 49170:REM ** GO TO ML S
     UBROUTINE
1280 FOR I=1 TO 2000:NEXT I
     :RETURN
     :REM ** TIMER FOR ERROR MES
     SAGES
1290 :
1300 DATA 232,3,1,8,2,8,19,8,8,
     192,240,193,12,0,0,0,0
1310 DATA 255,169,1,141,2,192,
     141,6,192,169,8,141,7,192,
     141,3,192
1320 DATA 141,5,192,169,2,141,4,
     192,169,0,141,12,192,32,179,
     193,238
1330 DATA 6,192,208,3,238,7,192,
     96,173,6,192,160,0,133,253,
     173,7
1340 DATA 192,133,254,138,145,
     253,96,32,50,192,32,4,193,
     162,44,32,59
1350 DATA 192,96,173,12,192,201,
     16,208,237,169,0,141,12,192,
     162,0,32
1360 DATA 50,192,32,59,192,32,50,
     192,32,66,193,32,170,193,32,
     251,192
1370 DATA 76,179,193,162,0,142,
     13,192,142,14,192,142,15,
     192,160,0,173
```

```
1380 DATA 8,192,133,253,173,9,
     192,133,254,177,253,201,100,
     208,7,169,0
1390 DATA 160,1,76,183,192,201,
     200,208,7,169,0,160,2,76,
     183,192,56
1400 DATA 200,233,100,144,5,240,
     29,76,169,192,136,105,100,
     140,13,192,192
1410 DATA 0,240,13,162,1,24,168,
     173,13,192,105,48,141,13,
     192,152,160
1420 DATA 0,24,56,233,10,200,144,
     5,240,6,76,206,192,136,105,
     10,141
1430 DATA 16,192,140,14,192,138,
     13,14,192,240,9,24,173,14,
     192,105,48
1440 DATA 141,14,192,173,16,192,
     24,105,48,141,15,192,96,238,
     8,192,208
1450 DATA 3,238,9,192,96,238,12,
     192,96,162,0,173,12,192,201,
     0,240
1460 DATA 26,32,50,192,32,59,192,
     32,50,192,32,59,192,32,66,
     193,162
1470 DATA 0,32,50,192,32,59,192,
     76,52,193,32,59,192,32,50,
     192,32
1480 DATA 59,192,32,50,192,173,6,
     192,133,45,173,7,192,133,46,
     0,173
1490 DATA 2,192,133,251,173,3,
     192,160,0,133,252,173,6,192,
     145,251,173
1500 DATA 4,192,133,251,173,5,
     192,133,252,173,7,192,145,
     251,174,6,192
1510 DATA 142,2,192,172,7,192,
     140,3,192,232,224,0,208,1,
     200,142,4
1520 DATA 192,140,5,192,96,173,8,
     192,205,10,192,208,11,173,9,
     192,205
1530 DATA 11,192,208,3,76,8,193,
     96,32,50,192,174,0,192,32,
     59,192
1540 DATA 32,50,192,174,1,192,32,
     59,192,32,50,192,162,131,32,
     59,192
1550 DATA 96,238,0,192,208,3,238,
     1,192,96,32,123,193,32,50,
     192,32
1560 DATA 143,193,32,122,192,174,
     13,192,224,0,240,6,32,50,
     192,32,59
1570 DATA 192,174,14,192,224,0,
     240,6,32,50,192,32,59,192,
     174,15,192
1580 DATA 32,50,192,32,59,192,32,
     87,192,32,251,192,32,123,
     193,76,188
1590 DATA 193,0,
1600 END
```

TRANS$8000

```
1000 REM ** TRANS$8000
1010 REM ** M.W. CAPRIO 1/5/84
1020 POKE 53281,6:POKE 53280,14
     :PRINT CHR$(154)
     :REM ** SET COLORS TO DEFAU
     LT VALUES
1030 PRINT"[CLEAR,SPACE4]
     WAIT WHILE THE BASIC LOADER
      RUNS"
1040 FOR I=32768 TO 33263:READ A
     :POKE I,A:NEXT
     :REM ** LOAD ML SUBROUTINE
1050 PRINT"[DOWN,SPACE8]
     PRESS RETURN TO BEGIN."
1060 GET C$:IF C$<>CHR$(13)THEN
     1060
1070 INPUT"[CLEAR]STARTING ADDRE
     SS (DECIMAL)";S
1080 INPUT"[DOWN]ENDING ADDRESS
     (DECIMAL)";E
1090 IF S<0 OR E<0 THEN PRINT"
     [DOWN]ILLEGAL QUANTITY ERRO
     R":GOSUB 1280:GOTO 1070
1100 IF S>65535 OR E>65535 THEN
     PRINT"[DOWN2]ILLEGAL QUANIT
     Y ERROR":GOSUB 1280
     :GOTO 1070
1110 IF S>=E THEN PRINT"[DOWN]
     END ADDRESS TOO LOW ERROR"
     :GOSUB 1280:GOTO 1070
1120 INPUT"[DOWN]IS THAT CORRECT
     ";C$:REM ** A     SECOND CH
     ANCE TO CORRECT ERRORS
1130 IF LEFT$(C$,1)<>"Y"THEN 1070
1140 PRINT"[DOWN]WITH WHAT NUMBE
     R DO YOU WANT YOUR"
1150 INPUT"DATA STATEMENTS TO BE
     GIN";D
1160 IF D<0 OR D>65535 THEN PRIN
     T"[DOWN]ILLEGAL QUANTITY ER
     ROR":GOSUB 1280:GOTO 1070
1170 :
1180 REM ** COMPUTE HIGH AND LOW
      HEX E-              QUIVALE
     NTS OF THE VARIABLES
1190 SH=INT(S/256):SL=S-SH*256
1200 E=E+1:EH=INT(E/256)
     :EL=E-EH*256
1210 DH=INT(D/256):DL=D-256*DH
1220 :
1230 REM ** PASS STARTING AND EN
     DING                ADDRESS
     ES AND DATA STATE-
1240 REM    MENT NUMBER TO ML SU
     BROUTINE
1250 POKE 32776,SL:POKE 32777,SH
     :POKE 32778,EL:POKE 32779,EH
1260 POKE 32768,DL:POKE 32769,DH
1270 SYS 32786:REM ** GO TO ML S
     UBROUTINE
1280 FOR I=1 TO 2000:NEXT I
     :RETURN
     :REM ** DELAY FOR ERROR MES
     SAGES
1290 :
1300 DATA 255,255,255,255,255,
     255,255,255,255,255,255,255,
     255,255,255,255,255
1310 DATA 255,169,1,141,2,128,
     141,6,128,169,8,141,7,128,
     141,3,128
1320 DATA 141,5,128,169,2,141,4,
     128,169,0,141,12,128,32,179,
     129,238
1330 DATA 6,128,208,3,238,7,128,
     96,173,6,128,160,0,133,253,
     173,7
1340 DATA 128,133,254,138,145,
     253,96,32,50,128,32,4,129,
     162,44,32,59
1350 DATA 128,96,173,12,128,201,
     16,208,237,169,0,141,12,128,
     162,0,32
1360 DATA 50,128,32,59,128,32,50,
     128,32,66,129,32,170,129,32,
     251,128
1370 DATA 76,179,129,162,0,142,
     13,128,142,14,128,142,15,
```

```
     128,160,0,173
1380 DATA 8,128,133,253,173,9,
     128,133,254,177,253,201,100,
     208,7,169,0
1390 DATA 160,1,76,183,128,201,
     200,208,7,169,0,160,2,76,
     183,128,56
1400 DATA 200,233,100,144,5,240,
     29,76,169,128,136,105,100,
     140,13,128,192
1410 DATA 0,240,13,162,1,24,168,
     173,13,128,105,48,141,13,
     128,152,160
1420 DATA 0,24,56,233,10,200,144,
     5,240,6,76,206,128,136,105,
     10,141
1430 DATA 16,128,140,14,128,138,
     13,14,128,240,9,24,173,14,
     128,105,48
1440 DATA 141,14,128,173,16,128,
     24,105,48,141,15,128,96,238,
     8,128,208
1450 DATA 3,238,9,128,96,238,12,
     128,96,162,0,173,12,128,201,
     0,240
1460 DATA 26,32,50,128,32,59,128,
     32,50,128,32,59,128,32,66,
     129,162
1470 DATA 0,32,50,128,32,59,128,
     76,52,129,32,59,128,32,50,
     128,32
1480 DATA 59,128,32,50,128,173,6,
     128,133,45,173,7,128,133,46,
     0,173
1490 DATA 2,128,133,251,173,3,
     128,160,0,133,252,173,6,128,
     145,251,173
1500 DATA 4,128,133,251,173,5,
     128,133,252,173,7,128,145,
     251,174,6,128
1510 DATA 142,2,128,172,7,128,
     140,3,128,232,224,0,208,1,
     200,142,4
1520 DATA 128,140,5,128,96,173,8,
     128,205,10,128,208,11,173,9,
     128,205
1530 DATA 11,128,208,3,76,8,129,
     96,32,50,128,174,0,128,32,
     59,128
1540 DATA 32,50,128,174,1,128,32,
     59,128,32,50,128,162,131,32,
     59,128
1550 DATA 96,238,0,128,208,3,238,
     1,128,96,32,123,129,32,50,
     128,32
1560 DATA 143,129,32,122,128,174,
     13,128,224,0,240,6,32,50,
     128,32,59
1570 DATA 128,174,14,128,224,0,
     240,6,32,50,128,32,59,128,
     174,15,128
1580 DATA 32,50,128,32,59,128,32,
     87,128,32,251,128,32,123,
     129,76,188
1590 DATA 129,0,
1600 END
```

Random Thoughts
Part 5: The Poisson Distribution

by Mark Zimmermann

In earlier articles we explored the idea of a random number distribution, a systematic pattern of probabilities which governs the chance of getting any particular number. We defined the moments of a distribution which help describe its shape. We also looked at the Gaussian, or "normal", distribution. This time, we're going to investigate another very important and useful random number distribution.

The Poisson distribution is named after a French mathematician; it's pronounced something like "pwa-SO". It comes up in many problems such as radioactive decay, where events occur somewhat rarely or spread out over time.

We'll look at one specific example below. It will lead us to the general formula for the Poisson random number distribution. If you don't care to read the details, you can skip ahead to the next section, where the results will be summarized and generalized further.

The Bridge

Suppose you are interested in the flow of traffic across a bridge. Cars arrive randomly at an average rate of one every 20 seconds. Each car takes ten seconds to cross the span. The bridge will be overloaded and suffer structural damage if three or more cars are on it at one time. Question: what fraction of the time do you expect the bridge to be overloaded?

We could stop right now and write up a computer program to simulate this situation. In fact, that's a lot of fun to do with animation and sound effects; if anybody produces something cute along those lines, please let me know!

Instead of a car-by-car simulation, however, let's try another approach. We can mathematically analyze the situation and get an answer for all values of the average time between cars, for various sizes and strengths of bridges. A literal simulation would only give us an answer for one specific choice of values. In fact, the simulation would only give us some sample values for the time the span is overloaded; we would have to run it many times and average the outcomes in order to estimate the actual answer. Analysis gives us a more direct solution in exchange for a little more thought.

Start out the analysis by defining some symbols for the critical numbers involved. Let X be the average time between cars (20 seconds), Y be the time a car takes to cross the bridge (ten seconds), and let Z be the number of cars needed to cause the bridge damage (three). Define the probability function P(N) to be the probability that N cars are crossing at one time. That is, P(0) is the chance that the bridge is unoccupied, P(1) is the chance for a single car to be crossing, etc.

The answer to our question, "What fraction of time is the bridge overloaded?" is just $P(3)+P(4)+\ldots$, where the "..." means "keep going up to infinity". From another point of view, the sum $P(3)+P(4)+\ldots$ is the chance that a snapshot, taken at a random moment of time, will reveal three or more cars on the bridge. Think about it for awhile, and you'll see that this "snapshot probability" is the same as the fraction of time that the bridge is overloaded. It's a nontrivial point, so if it seems obvious, think again.

Since there must always be some number of cars on the bridge at any one time, the sum $P(0)+P(1)+\ldots$ must add up to one (certainty, in probability language). Juggling the P's to get the answer to our problem on one side of the equation shows that the answer is: $P(\text{damage}) = P(3)+P(4)+\ldots = 1-P(0)-P(1)-P(2)$. Both ways of writing the answer are equivalent, but the second way only involves a finite number of operations and may turn out to be easier to handle when we want to plug some numbers in.

What is P(N)? We're going to have to sneak up on the answer. Cars can arrive at any time, so that seems to be a continuous random process; on the other hand, there are always an integer number of cars on the bridge at once, so that seems to be a discrete random process. We'll have to be clever to get the solution to this hybrid question. (That answer will be the

Poisson distribution.)

Begin with P(0). How can there be no cars on the bridge? There must not have been any cars arriving for the previous Y seconds. Take our specific case with Y=10. A car arrives on the average once every X=20 seconds; that means that, on the average, there is half a car every ten seconds. Half a car makes no sense, but maybe Y/X = 0.5 is not a bad guess for P(0)?

We can do better. Consider the situation for the ten seconds before our snapshot is taken, one second at a time. At each of those seconds, we can estimate the chance of a car appearing as: 1/X = 1/20 = 0.05. That is, only 5% of the time (roughly) is there a car entering the scene within any given second. So: 1−1/X = 1−0.05 = 0.95 = 95% of the time, there is no car. If each of the seconds is an independent test, then for there to be no car ten times in a row we have to have ten sequential non-events, each with probability 0.95. The chance of that happening is: 0.95*0.95*... *0.95 (ten times) = 0.95↑10 = 5987.... So, is P(0) = 0.5987?

Again, we can do better. Why did our latest estimate of P(0) get a bit bigger result than our first guess, where we took the whole ten-second interval as one unit? Answer: because the first way effectively ignored the chance that more than one car might come along within the time period. (The arrival rate of single cars is lower, because when multiple cars arrive they pull up the average and allow us to get by with fewer single cars.) The weakness of our latest estimate, where we broke the ten-second interval into ten parts, is that there again we ignored the chance that more than one car might appear during any given time period (one second). The chance of two or more cars within a second is much smaller than the chance of two or more within a ten-second period, so our error was much smaller the latest time.

We could get very close to an exact answer by dividing the critical ten-second period into more sub-parts. (In fact, if you remember from Part 3 of "Random Thoughts", this subdividing is just like what we did to determine the area under a curve... it's really a calculus trick, only we're not admitting that you know calculus!) Suppose the interval, Y seconds long, is broken into M parts. The chance of a car's arrival during one of those M parts is roughly Y/(M*X). That estimate gets more precise as M gets bigger. The chance of *no* car's arrival is, of course, 1−Y/(M*X). Multiplying up M of those probabilities together tells us that we should estimate (P(0) is about (1−Y/(M*X))↑M. We could now plug in successively larger values for M, and get successively more accurate estimates for P(0).

You can try this, if you like. Your estimates of P(0) should rapidly converge to 0.60653.... If you're a suspicious sort, you may want to try taking the natural logarithm of 0.60653.... You'll find that, in BASIC, LOG (0.60653) is very close to −0.5. That's no coincidence! In the limit, as M gets very large, (1−Y/(M*X))↑M becomes EXP(−Y/X) = EXP(−0.5) for our values of Y and X. That's one of the ways to define the EXP (exponential) function. So we've discovered that P(0) = EXP(−Y/X).

From here on, it will go quicker. What is P(1), the chance for one car to be on the bridge when we take our snapshot? Again breaking the time interval Y up into M intervals, we want to ask, "How likely is it for there to be *no* car in M-1 intervals, and *one* car in one interval? With cars coming at rate X, the chance of one car in an interval of length Y/M is roughly (Y/M)/X = Y/(M*X). The chance of no car is 1−Y/(M*X). Multiplying up the probabilities for each interval, the chance of one car in the first interval and no cars in any other interval is just (Y/(M*X))*(1−Y/(M*X))↑(M−1).

But there are M intervals during which the car could have come, so we have to multiply the above result by M, since it only counted the chance that the car arrived during the first interval. The result is an estimate: P(1) is about (Y/X)*(1−Y/(M*X))↑(M−1). Letting M get large, we discover that this converges to simply P(1) = (Y/X)*EXP(−Y/X) = 0.30327... for our specific case.

We're almost there now! All

that's left to figure is P(2). By similar methods to what we've just done, you can discover that P(2) is approximately (M*(M−1)/ 2)* (Y/(M*X))↑2 * (1−Y/(M*X))↑ (M−2) for large enough M. The M*(M−1)/2 comes from the various ways to choose two intervals out of M, where we don't care about the order they are chosen in. The (Y/(M*X))↑2 comes from multiplying together the chance for a single car in each of two intervals. And the rest of the expression is the result of multiplying together the chance for no car in M−2 intervals. In the limit of large M, this formula reduces to P(2) = (1/2) * (Y/X)↑2 * EXP(−Y/X) = 0.07582 for our specific X and Y.

So there's our answer: the bridge is overloaded 1−0.60653 −0.30327−0.07582 = 0.01439, that is, about 1.4% of the time. How close did your simulation come to that result? Table 1 summarizes what we've found in solving the bridge problem. I've included the general formula for P(N); you can check that it reduces to the cases we derived above for N = 0, 1, and 2. (Note that N!, read "N factorial", is just 1*2*3*... *N, and 0! is defined as 1.)

Table 1

P(N) = (Y/X)↑N * EXP(−Y/X) / N!

N	*P(N)*
0	0.60653
1	0.30327
2	0.07582
3	0.01264
4	0.00158
5	0.00016
6	0.00001

The Next Section

As promised, here is The Next Section. We've finished a long but rather interesting example. What have we learned?

We've seen the distribution P(N) = A↑N * EXP(−A) / N! where A is the average rate of events and N is the number we are asking about. In the bridge example, A was just Y/X, the average rate of cars arriving during a ten-second interval. Exactly the same distribution P(N) will work for calculating the chance of having N telephone calls arriving at a switchboard at once, N radioactive atoms decaying within a chosen time interval, etc. In fact, the Poisson distribution applies to a host of circumstances.

The key thing to remember about the Poisson distribution is that the events in question must be *independent,* not correlated or linked with each other. If cars travel in caravans, for instance, our bridge will be overloaded more often than we estimated. Contrariwise, if cars avoid each other and spread themselves out, the bridge will rarely if ever be overloaded, and our estimate is too high. If a rash of telephone calls are provoked by some natural disaster, they won't be independent of each other. If the radioactive atoms in question are part of a nuclear reactor and promote each other's decay, they aren't independent either.

The Poisson distribution is quite spread out. Its mean value (or "first moment") is, by definition, A in the formula. Its standard deviation turns out also to be A. For small values of A, the Poisson distribution piles up near zero, as we saw in Table 1. As A gets larger, however, the distribution approaches the normal Gaussian distribution we saw in Part 4 of this series. We should have expected that... the Gaussian distribution is what almost everything tends toward, eventually.

Next time: Random Walks! **C**

250-Digit Arithmetic

by William Miller

Have you ever wished that your computer could do arithmetic involving numbers longer than the eight-to-twelve digit maximum? Here is a series of programs I wrote to do just that. These programs add, subtract, multiply, divide, raise to a power and find the square root.

To perform these tasks I treat the numbers as strings and then manipulate the strings. I use the computer in two ways—as a calculator and as a worksheet. The processes mimic the ones we use when we perform arithmetic using our brain and a sheet of paper. They incorporate various routines to perform (1) number (string) input, (2) deletions, (3) decimal point location and deletion, (4) zero suppression, (5) rounding off, (6) formatting process inputs and (7) formatting outputs.

For all but the addition and the subtraction programs I use the Trachtenberg system of basic math. This system is easier, quicker and more sophisticated than the traditional system taught in school. It is also more compatible with computer logic. These programs can be used for the purposes of arithmetic, logic and education. That is, they can give an answer to an equation, can demonstrate the uses of various procedures and also allow you to learn the Trachtenberg system. If you have the desire and the computer memory you can combine and expand these programs.

For the purposes of education, debugging and monitoring the calculations, additional statements can be added. These may be remarks within the program or may be print statements to display messages, intermediate results, counter values or progress (location) within the program. For example, 551 PRINT "R$-551-";R$ shows the result string (R$) at location 551. The display of the location gives a more obvious trace and allows for easier removal of the statement.

Addition This is a straightforward representation of common addition. The strings are aligned on their (implied) decimal points, zeros are added to produce equal lengths, addition is performed (and repeated, if so specified) and the result is formatted.

Subtraction This is like addition; however, a sign is printed and only one calculation is performed.

Multiplication This is like addition; however, the input is zero suppressed and the process follows the Trachtenberg system. In this system a UT sum (or units-tens) is a basic element and is created by "multiplying" a pair of numbers (from a source such as a multiplicand) by another number (from a multiplier, for example). For instance:

U T
45×7→28+35→11 (the UT)
where 28=4×7 and 35=5×7

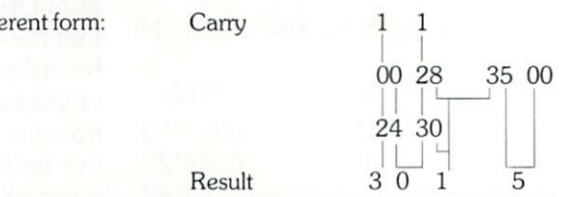
The units and tens digits are added to form the UT (pair) product.

Multiplication of two numbers requires the addition of UT products, as in the following example:

45×67=3015

STEP	UT PRODUCT	RESULT	CARRY
(1) 5(0)×7→35+00	5	5	0
(2) 45×7→28+35	11		
5(0)×6→30+00	0	1	1
(3) (0)4×7→00+28	2		
45×6→24+30	7	0	1
(4) (0)4×6→00+24	2	3	0

or in different form:



For the computer program, only two numbers are multiplied. The units digits are added for one sum (S), and the tens units are added for another sum (CC). The "result" is the units digit of the sum of S, CC, and the carry (SC) of the previous result.

Division This is like multiplication as far as using the same input routines and the Trachtenberg system. The numbers are compared in order to determine the length of the result. After zero fill to initialize, the strings are processed. In addition to the UT sum found in multiplication three is also the NT (number-tens) sum. Thus: 45×7→28+35→31 (NT).

The first digit of the dividend is brought down to form the partial dividend (A2), which is then divided by the first digit of the divisor to give the trial number (X). The NT sum of this (X) number times the first pair (from the left) of the divisor gives S2, which is subtracted from A2 to give the carry (C). C is concatenated to the next digit of the dividend to form the working figure (A1). The UT sums are added (S1) and subtracted from A1 to give the next partial dividend (A2). Whenever A2 or C is negative, X is adjusted and the process repeated.

The UT pairs are initiated at the second position of the divisor (proceeding to the right) and the last position of the result string (to left) as already created. For example:

789999÷4567=172.979855

A1	7	38	29	59	59	69	60	50	50	20
S1	0	5	15	14	22	23	19	23	26	12
A2	7	33	14	45	37	46	41	27	24	8
S2	4	31	9	40	31	40	36	22	22	
C	3	2	5	5	6	6	5	5	2	
						↑				

Note: S1 (45+54)+(12+14)+(49+00)→22
as (56×9)+(67×2)+(7(0)×7)
S2 (28+35)→31 (NT) as (45×7)

In the program, the size of the array (line 40) that is allowable depends upon the memory available. VIC users can use an integer variable (ex, 0%(I)) to conserve memory.

Power This is like multiplication, but the process is repeated according to the power specified.

Square Root This is like division in using cross multiplication. The input is formatted and the first two digits are brought down for a trial quotient (A1). The largest square root in this becomes the first digit (trial result) of the result string. Its square is subtracted from A1 to give the trial difference (AT). After testing to make sure that it is not greater than the trial quotient (A1) or greater than twice the result (so far) it is used to create a test dividend. This is five times the two-digit number beginning with the second position of AT.

This test dividend is divided by the trial result (R) to give the test number (X), which is "cross multiplied" by the result string to give X1. AT is concatenated with the next two digits of the input string, from which X1 is subtracted, giving the new trial difference (AT). As stated, if AT is too large or too small, the test number (X) is adjusted and the process repeated.

The "cross products" in X1 are summed as in this example: R$=29, X=5

```
  20                  where 2×(5×2)→20
  90                        2×(5×9)→90
   25                         (5×5)→25
 2925
```

$\sqrt{8724321}$ = 2953.696159

A1	08	0472	03143	021821	0411200	05680400
X1	04	0441	02925	017709	0354396	05316561
AT	04	0031	00218	004112	0056804	00363839
C1	04	03	02	04	05	03
C2	20	15	10	20	25	15
test	9	7-5	5-3	9-6	9	7-6
ans.	29	295	2953	29536	295369	2953696

Additional Comments Although these programs can be crunched even more, I left them open for easier understanding. Besides crunching and combining, they may be expanded by processing and array of strings. Referencing a string could be done by using a counter and subscripts (for example, I C/250). Extreme precision is possible; however, one should consider the factors of significant digits, practicality and running time. Math can be very precise, but en-

gineering and statistics may not need such precision considering their types of inputs and outputs.

Besides using strings and the Trachtenberg system, there is another trick involving multiplication that I happened to discover. By breaking a number down into groups (strings) and manipulating these groups as if they were single digits, precision can be increased by several times. It also works with calculators. Groups are multiplied by each other, and their products are added together on paper.

For example:

99999999 × 99999999 = 9999999800000001

(1) by groups of four

```
 9999      9999      ×      9999      9999
 A         B                C         D

        99980001                        ______ B × D
     99980001                           ______ B × C
     99980001                           ______ A × D
  99980001                              ______ A × C
  ---------------------------------
  9999999800000001
```

(2) by groups of three

```
 99    999    999      ×      99    999    999
 A     B      C                D     E      F

          998001                        ______ C × F
        998001                          ______ C × E
      98901                             ______ C × D
        998001                          ______ B × F
     998001                             ______ B × E
   98901                                ______ B × D
      98901                             ______ A × F
   98901                                ______ A × E
  9801                                  ______ A × D
  ---------------------------------
  9999999800000001
```

C

References

Cutler, Ann and Rudolf McShane. *The Trachtenberg Speed System of Basic Mathematics.* Doubleday & Co. (Garden City, New York: 1960).

Addition Crunch

```
10 PRINT"250-DIGIT ADDITION"
20 PRINT "ENTER EQUATION"
30 D$=".":GOSUB 1000:A$=C$:AL=CL
40 PRINT:GOSUB 1000:B$=C$:BL=CL
50 F=ABS(AL-BL):IF AL>=BL THEN 70
60 Z$=A$:A$=B$:B$=Z$:AL=BL
70 F$="0":IF F=0 THEN 90
80 FOR I=1 TO F:B$=F$+B$:NEXT I
90 LA=LEN(A$):LB=LEN(B$)
   :F=ABS(LA-LB):IF LA>LB THEN 1
   20
100 IF LA=LB THEN 130
110 FOR I=1 TO F:A$=A$+F$:NEXT I
    :GOTO 130
120 FOR I=1 TO F:B$=B$+F$:NEXT I
130 LA=LEN(A$):C=0:R$=""
140 FOR I=LA TO 1 STEP -1
    :A1$=MID$(A$,I,1)
    :B1$=MID$(B$,I,1)
150 A1=VAL(A1$):B1=VAL(B1$)
    :D1=A1+B1+C:C=0
    :IF D1<10 THEN 170
160 D1=D1-10:C=1
170 D1$=STR$(D1):D1$=RIGHT$(D1$,
    1):R$=D1$+R$
180 NEXT I:LR=LA-AL
    :IF C=0 THEN 200
190 C$=STR$(C):C$=RIGHT$(C$,1)
    :R$=C$+R$:AL=AL+1
200 IF CC$<>"+" THEN 220
210 A$=R$:GOTO 40
220 IF LR=0 THEN 270
230 IF AL>0 THEN 250
240 RL$="0":GOTO 260
250 RL$=LEFT$(R$,AL)
260 RR$=RIGHT$(R$,LR)
    :R$=RL$+D$+RR$
270 PRINT :PRINT R$:END
1000 C$=""
1010 GET CC$:IF CC$="" THEN 1010
1020 PRINT CC$;:LC=LEN(C$)
     :IF CC$="+" OR CC$="=" THEN
      1060
1030 IF CC$<>CHR$(20) THEN 1050
1040 C$=LEFT$(C$,LC-1)
     :PRINT"[LEFT,RIGHT]";
     :GOTO 1010
1050 C$=C$+CC$:GOTO 1010
1060 CL=LC:CR=0:FOR I=1 TO LC
     : CH$=MID$(C$,I,1)
     :IF CH$<>D$ THEN 1080
1070 CL=I-1:CR=LC-I
1080 NEXT I:CL$=LEFT$(C$,CL)
     :CR$=RIGHT$(C$,CR)
     :C$=CL$+CR$
1090 RETURN
```

Subtraction Crunch

```
10 PRINT"250-DIGIT SUBTRACTION"
20 PRINT "ENTER EQUATION"
30 D$=".":S$="":GOSUB 1000:A$=C$
   :AL=CL
40 PRINT:GOSUB 1000:B$=C$:BL=CL
50 F=ABS(AL-BL):IF AL>=BL THEN 70
60 Z$=A$:A$=B$:B$=Z$:AL=BL:S$="-"
70 F$="0":IF F=0 THEN 90
80 FOR I=1 TO F:B$=F$+B$:NEXT I
90 LA=LEN(A$):LB=LEN(B$)
   :F=ABS(LA-LB):IF LA>LB THEN 1
   20
100 IF LA=LB THEN 130
110 FOR I=1 TO F:A$=A$+F$:NEXT I
    :GOTO 130
120 FOR I=1 TO F:B$=B$+F$:NEXT I
130 LA=LEN(A$):C=0:R$=""
140 FOR I=LA TO 1 STEP -1
    :A1$=MID$(A$,I,1)
    :B1$=MID$(B$,I,1)
150 A1=VAL(A1$)-C:B1=VAL(B1$):C=0
    :IF A1>=B1 THEN 170
160 A1=A1+10:C=1
170 D1=A1-B1:D1$=STR$(D1)
```

```
    :D1$=RIGHT$(D1$,1):R$=D1$+R$
180 NEXT I:IF C=0 THEN 200
190 Z$=A$:A$=B$:B$=Z$:S$="-"
    :GOTO 130
200 LR=LA-AL:IF AL>0 THEN 220
210 RL$="0":GOTO 290
220 IF AL=1 THEN 280
230 FOR I=1 TO AL-1
    :CH$=MID$(R$,I,1)
    :IF CH$<>F$ THEN 250
240 NEXT I
250 IF LR>0 THEN 270
260 D$=""
270 AL=AL-I+1
280 LT=AL+LR:R$=RIGHT$(R$,LT)
    :RL$=LEFT$(R$,AL)
290 RR$=RIGHT$(R$,LR)
    :R$=S$+RL$+D$+RR$
300 PRINT:PRINT R$:END
```

```
1000 C$=""
1010 GET CC$:IF CC$="" THEN 1010
1020 PRINT CC$;:LC=LEN(C$)
     :IF CC$="-" OR CC$="=" THEN
      1060
1030 IF CC$<>CHR$(20) THEN 1050
1040 C$=LEFT$(C$,LC-1)
     :PRINT"[LEFT,RIGHT]";
     :GOTO 1010
1050 C$=C$+CC$:GOTO 1010
1060 CL=LC:CR=0:FOR I=1 TO LC
1070 CH$=MID$(C$,I,1)
     :IF CH$<>D$ THEN 1090
1080 CL=I-1:CR=LC-I
1090 NEXT I:CL$=LEFT$(C$,CL)
     :CR$=RIGHT$(C$,CR)
     :C$=CL$+CR$
1100 RETURN
```

Multiplication Crunch

```
10 PRINT "250-DIGIT MULTIPLICATI
   ON"
20 PRINT "ENTER EQUATION"
30 D$=".":F$="0"
40 GOSUB 1000:A$=C$:AL=CL
   :LA=LEN(A$)
50 PRINT:GOSUB 1000:B$=C$:BL=CL
   :LB=LEN(B$)
60 FOR I=1 TO LA:B$=F$+B$:NEXT I
70 FOR I=1 TO LB:A$=F$+A$:NEXT I
80 LT=LA+LB:IF LT<=250 THEN 100
90 PRINT "NUMBERS TOO BIG"
   :GOTO 20
100 C=0:SC=0:R$=""
110 FOR I=1 TO LT:N=LT-I+1:S=0
    :CC=0
120 FOR J=1 TO I:X=N+J-1:Y=LT-J+1
130 AD$=MID$(A$,X,1)
    :BD$=MID$(B$,Y,1)
140 AD=VAL(AD$):BD=VAL(BD$)
150 XX=AD*BD:XL=0:XR=XX
    :IF XX<10 THEN 180
```

```
160 XX$=STR$(XX):XL$=MID$(XX$,2,
    1):XR$=RIGHT$(XX$,1)
170 XL=VAL(XL$):XR=VAL(XR$)
180 CC=CC+XL:S=S+XR
190 NEXT J:S=S+C+SC:C=CC
200 S$=STR$(S):SR$=RIGHT$(S$,1)
    :SR=VAL(SR$)
210 SC=(S-SR)/10:R$=SR$+R$
220 NEXT I:RL=AL+BL
    :CH$=LEFT$(R$,1)
    :IF CH$<>F$ THEN 240
230 LT=LT-1:RL=RL-1
    :R$=RIGHT$(R$,LT)
240 IF CC$<>"*" THEN 260
250 LA=LT:AL=RL:A$=R$:GOTO 50
260 RR=LT-RL:IF RL>0 THEN 300
270 IF RL=0 THEN 290
280 F=ABS(RL):FOR I=1 TO F
    :R$=F$+R$:NEXT I
290 R$=F$+D$+R$:GOTO 330
300 IF RR>0 THEN 320
310 D$=""
```

```
320 RL$=LEFT$(R$,RL)
    :RR$=RIGHT$(R$,RR):R$=RL$+RR$
330 PRINT:PRINT R$:END
1000 C$=""
1010 GET CC$:IF CC$="" THEN 1010
1020 PRINT CC$;:LC=LEN(C$)
     :IF CC$="*" OR CC$="=" THEN
      1060
1030 IF CC$<>CHR$(20) THEN 1050
1040 C$=LEFT$(C$,LC-1)
     :PRINT "[LEFT,RIGHT]";
     :GOTO 1010
1050 C$=C$+CC$:GOTO 1010
1060 CL=LC:CR=0:FOR I=1 TO LC
     :CH$=MID$(C$,I,1)
     :IF CH$<>D$ THEN 1080
1070 CL=I-1:CR=LC-I
1080 NEXT I:CL$=LEFT$(C$,CL)
     :CR$=RIGHT$(C$,CR)
     :C$=CL$+CR$
1090 LC=LEN(C$):FOR I=1 TO LC
     :CH$=MID$(C$,I,1)
     :IF CH$<>F$ THEN 1110
1100 NEXT I
1110 C$=RIGHT$(C$,LC-I+1)
     :CL=CL-I+1
1120 CL$="":CR$="":RETURN
```

Division Crunch

```
10 PRINT "250-DIGIT DIVISION"
20 PRINT"ENTER LENGTH OF ANSWER"
   :INPUT LX
30 PRINT "ENTER EQUATION"
40 D$=".":F$="0":DIM C%(100)
50 GOSUB 1000:A$=C$:AL=CL:PRINT
60 GOSUB 1000:B$=C$:BL=CL:C$=""
70 LA=LEN(A$):LB=LEN(B$):AA$=A$
   :BB$=B$:F=ABS(LA-LB)
80 IF LA>LB THEN 110
90 IF LA=LB THEN 120
100 FOR I=1 TO F:AA$=AA$+F$
    :NEXT I:GOTO 120
110 FOR I=1 TO F:BB$=BB$+F$
    :NEXT I
120 LS=LEN(AA$):C=0
130 FOR I=LS TO 1 STEP -1
    :A1$=MID$(AA$,I,1)
    :B1$=MID$(BB$,I,1)
140 A1=VAL(A1$)-C:C=0:B1=VAL(B1$)
    :IF A1>=B1 THEN 160
150 C=1
160 NEXT I:AA$="":BB$=""
    :LR=AL-BL+1:CL=1
    :IF C=0 THEN 180
170 LR=LR-1:CL=2
180 IF LR>=LX THEN 200
190 F=LX-LR+CL+3:FOR I=1 TO F
    :A$=A$+F$:NEXT I
200 B$=B$+F$:R$="":FOR I=1 TO LB
    -1:R$=F$+R$:NEXT I
210 LX=LX+1:A2$=LEFT$(A$,1)
    :A2=VAL(A2$):C(1)=A2
220 FOR I=2 TO LX+CL:S1=0
    :B1$=MID$(B$,1,1)
    :B2$=MID$(B$,2,1)
230 B1=VAL(B1$):B2=VAL(B2$)
    :X=INT(A2/B1):IF X<10 THEN 2
    50
240 X=9
250 X1=X*B1:X2=X*B2:X2$=STR$(X2)
    :XR$=RIGHT$(X2$,1)
    :XR=VAL(XR$)
260 X2=(X2-XR)/10:S2=X1+X2
    :C=A2-S2:IF C<0 THEN 350
270 A1$=MID$(A$,I,1)
    :A1=VAL(A1$)+(10*C)
    :BL$=MID$(B$,2,1)
280 BR$=MID$(B$,3,1):BL=VAL(BL$)
    :BR=VAL(BR$):XX=X
290 GOSUB 2000:S1=RL+RR
300 FOR J=3 TO LB:JR=J+1
    :LD=LEN(R$):RJ=LD-J+3
310 R1$=MID$(R$,RJ,1)
```

```
    :BL$=MID$(B$,J,1)
    :BR$=MID$(B$,JR,1)
320 XX=VAL(R1$):BL=VAL(BL$)
    :BR=VAL(BR$)
330 GOSUB 2000:S1=S1+RL+RR
340 NEXT J:IF A1>=S1 THEN 390
350 IF X=0 THEN 370
360 X=X-1:GOTO 250
370 I=I-1:CH$=MID$(R$,LD,1)
    :LD=LD-1:IF CH$=F$ THEN 370
380 X=VAL(CH$)-1:R$=LEFT$(R$,LD)
    :A2=C%(I-1):GOTO 250
390 A2=A1-S1:C%(I)=A2:X1$=STR$(X)
    :X1$=RIGHT$(X1$,1):R$=R$+X1$
400 NEXT I:R$=RIGHT$(R$,LX):CO=5
    :A1$=""
410 FOR I=LX TO 1 STEP-1
    :R1$=MID$(R$,I,1)
420 R1=VAL(R1$)+CO:CO=0
    :IF R1<10 THEN 440
430 R1=R1-10:CO=1
440 R1$=STR$(R1):R1$=RIGHT$(R1$,
    1):A1$=R1$+A1$
450 NEXT I:LX=LX-1:R$=LEFT$(A1$,
    LX):IF CO=0 THEN 470
460 CO$=STR$(CO):CO$=RIGHT$(CO$,
    1):R$=CO$+R$:LR=LR+1
470 IF LR>0 THEN 510
480 IF LR=0 THEN 500
490 F=ABS(LR):FOR I=1 TO F
    :R$=F$+R$:NEXT I
500 R$=F$+D$+R$:GOTO 550
510 RR=ABS(LX-LR):IF LX>LR THEN
    540
520 IF LX=LR THEN 550
530 FOR I=1 TO RR:R$=R$+F$:NEXT I
    :GOTO 550
540 RL$=LEFT$(R$,LR)
    :RR$=RIGHT$(R$,RR)
    :R$=RL$+D$+RR$
550 PRINT:PRINT R$:END
1000 C$=""
1010 GET CC$:IF CC$="" THEN 1010
1020 PRINT CC$;:LC=LEN(C$)
    :IF CC$="/" OR CC$="=" THEN
     1060
1030 IF CC$<>CHR$(20) THEN 1050
1040 C$=LEFT$(C$,LC-1)
    :PRINT "[LEFT,RIGHT]";
    :GOTO 1010
1050 C$=C$+CC$:GOTO 1010
1060 CL=LC:CR=0:FOR I=1 TO LC
    :CH$=MID$(C$,I,1)
    :IF CH$<>D$ THEN 1080
1070 CL=I-1:CR=LC-I
1080 NEXT I:CL$=LEFT$(C$,CL)
    :CR$=RIGHT$(C$,CR)
    :C$=CL$+CR$
1090 LC=LEN(C$):FOR I= 1 TO LC
    :CH$=MID$(C$,I,1)
    :IF CH$<>F$ THEN 1110
1100 NEXT I
1110 C$=RIGHT$(C$,LC-I+1)
    :CL=CL-I+1
1120 CL$="":CR$="":RETURN
2000 RL=XX*BL:RL$=STR$(RL)
    :RL$=RIGHT$(RL$,1)
    :RL=VAL(RL$)
2010 RR=XX*BR:RR$=STR$(RR)
    :RR$=RIGHT$(RR$,1)
    :RX=VAL(RR$)
2020 RR=(RR-RX)/10:RETURN
```

Power Crunch

```
10 PRINT "POWER"
20 PRINT "ENTER EQUATION":A$=""
30 GET AA$:IF AA$="" THEN 30
40 PRINT AA$;:LA=LEN(A$)
   :IF AA$="^" THEN 80
50 IF AA$<>CHR$(20) THEN 70
```

```
60 A$=LEFT$(A$,LA-1)
   :PRINT "[LEFT,RIGHT]";:GOTO 30
70 A$=A$+AA$:GOTO 30
80 AL=LA:F$="0":D$="."
90 FOR I=1 TO LA:CH$=MID$(A$,I,1)
   :IF CH$<>D$ THEN 110
100 AL=I-1:AR=LA-I
110 NEXT I:AL$=LEFT$(A$,AL)
    :AR$=RIGHT$(A$,AR):A$=AL$+AR$
    :PRINT
120 GET PP$:IF PP$="" THEN 120
130 PRINT PP$;:LP=LEN(P$)
    :IF PP$="=" THEN 170
140 IF PP$<>CHR$(20) THEN 160
150 P$=LEFT$(P$,LP-1)
    :PRINT "[LEFT,RIGHT]";
    :GOTO 120
160 P$=P$+PP$:GOTO 120
170 P=VAL(P$):LA=LEN(A$):LR=LA
    :BL=AL:C$=A$
180 FOR I=1 TO P-1:B$=C$
    :FOR J=1 TO LR:B$=F$+B$
    :NEXT J
190 FOR J=1 TO LA:A$=F$+A$:NEXT J
200 C=0:SC=0:R$="":LT=LR+LA
210 FOR J=1 TO LT:N=LT-J+1:S=0
    :CC=0
220 FOR K=1 TO J:X=N+K-1:Y=LT-K+1
230 AD$=MID$(A$,X,1)
    :BD$=MID$(B$,Y,1):AD=VAL(AD$)
    :BD=VAL(BD$)
240 XX=AD*BD:XL=0:XR=XX
    :IF XX<10 THEN 270
250 XX$=STR$(XX):XL$=MID$(XX$,2,
    1):XR$=RIGHT$(XX$,1)
260 XL=VAL(XL$):XR=VAL(XR$)
270 CC=CC+XL:S=S+XR
280 NEXT K:S=S+C+SC:C=CC
    :S$=STR$(S):SR$=RIGHT$(S$,1)
290 SR=VAL(SR$):SC=(S-SR)/10
    :R$=SR$+R$
300 NEXT J:RL=AL+BL
310 CH$=LEFT$(R$,1)
    :IF CH$<>F$ THEN 330
320 LT=LT-1:RL=RL-1
    :R$=RIGHT$(R$,LT)
330 LR=LT:AL=RL:A$=R$
340 NEXT I:RR=LT-RL
    :IF RL>0 THEN 380
350 IF RL=0 THEN 370
360 F=ABS(RL):FOR I=1 TO F
    :R$=F$+R$:NEXT I
370 R$=F$+D$+R$:GOTO 410
380 IF RR>0 THEN 400
390 D$=""
400 RL$=LEFT$(R$,RL)
    :RR$=RIGHT$(R$,RR)
    :R$=RL$+D$+RR$
410 PRINT:PRINT R$:END
```

Square Crunch

```
10 PRINT "SQUARE ROOT"
20 PRINT "ENTER LENGTH OF ANSWER"
   :INPUT LX
30 PRINT "ENTER NUMBER":A$=""
40 GET AA$:IF AA$="" THEN 40
50 PRINT AA$;:LA=LEN(A$)
   :IF AA$=CHR$(13) THEN 90
60 IF AA$<>CHR$(20) THEN 80
70 A$=LEFT$(A$,LA-1)
   :PRINT "[LEFT,RIGHT]";:GOTO 40
80 A$=A$+AA$:GOTO 40
90 AL=LA:F$="0":D$="."
100 FOR I=1 TO LA:CH$=MID$(A$,I,
    1):IF CH$<>D$ THEN 120
110 AL=I-1:AR=LA-I
120 NEXT I:AL$=LEFT$(A$,AL)
    :AR$=RIGHT$(A$,AR):A$=AL$+AR$
130 LA=LEN(A$):FOR I=1 TO LA
    :CH$=MID$(A$,I,1)
    :IF CH$<>F$ THEN 150
140 NEXT I
150 AL=AL-I+1:AL$="":AR$=""
    :IF AL<0 THEN 180
160 IF ((AL/2)-INT(AL/2))=0 THEN
```

```
    210
170 A$=F$+A$:GOTO 210
180 IF ((I/2)-INT(I/2))<>0 THEN
    200
190 I=I-1
200 A$=RIGHT$(A$,LA-I+1)
210 LA=LEN(A$):IF ((LA/2)-INT(LA
    /2))=0 THEN 230
220 A$=A$+F$:LA=LA+1
230 F=(LX*2)-LA:FOR I=1 TO F+4
    :A$=A$+F$:NEXT I
240 A1$=MID$(A$,1,2):A1=VAL(A1$)
    :R=INT(SQR(A1))
250 R$=STR$(R):R$=RIGHT$(R$,1)
    :C1=A1-(R*R):A2$=STR$(C1)
260 LA=LEN(A2$)-1:A2$=RIGHT$(A2$,
    LA):IF LA=2 THEN 280
270 A2$=F$+A2$
280 LX=2*LX:X=INT(5*C1/R)
    :IF X<10 THEN 300
290 X=9
300 FOR I=3 TO LX+4 STEP 2
310 X1=X*X:X1$=STR$(X1)
    :X1$=RIGHT$(X1$,1)
320 X2=VAL(X1$):CC=(X1-X2)/10
    :LR=LEN(R$)
330 FOR J=LR TO 1 STEP -1
    :X3$=MID$(R$,J,1):X3=VAL(X3$)
340 X3=2*(X*X3)+CC:X3$=STR$(X3)
    :X3$=RIGHT$(X3$,1)
350 X4=VAL(X3$):CC=(X3-X4)/10
    :X1$=X3$+X1$
360 NEXT J:IF CC=0 THEN 410
370 CC$=STR$(CC):LC=LEN(CC$)-1
    :CC$=RIGHT$(CC$,LC)
380 FOR J=1 TO LC:CH$=MID$(CC$,J,
    1):IF CH$<>F$ THEN 400
390 NEXT J
400 CC$=RIGHT$(CC$,LC-J+1)
    :X1$=CC$+X1$
410 L2=LEN(X1$):A1$=MID$(A$,I,2)
    :A1$=A2$+A1$
420 L1=LEN(A1$):L3=L1-L2
    :IF L1=L2 THEN 440
430 FOR J=1 TO L3:X1$=F$+X1$
    :NEXT J
440 CT=0:AT$="":FOR J=L1 TO 1 ST
    EP -1
450 A3$=MID$(A1$,J,1)
    :A4$=MID$(X1$,J,1)
    :A3=VAL(A3$)-CT
460 CT=0:A4=VAL(A4$)
    :IF A3>=A4 THEN 480
470 A3=A3+10:CT=1
480 A5=A3-A4:A5$=STR$(A5)
    :A5$=RIGHT$(A5$,1)
    :AT$=A5$+AT$
490 NEXT J:IF CT=0 THEN 510
500 X=X-1:GOTO 310
510 X$=STR$(X):X$=RIGHT$(X$,1)
    :X$=F$+F$+R$+X$
520 C5=0:C6=0:FOR J=L1 TO 1 STEP
    -1
530 A4$=MID$(X$,J,1):A4=VAL(A4$)
    :A5=(A4*2)+C5
540 C5=0:IF A5<10 THEN 560
550 A5=A5-10:C5=1
560 A5=A5-C6:C6=0:A3$=MID$(AT$,J,
    1)
570 A3=VAL(A3$):IF A5>=A3 THEN 5
    90
580 C6=1
590 NEXT J:IF C5>=C6 THEN 610
600 X=X+1:GOTO 310
610 AT$=RIGHT$(AT$,L1-1)
    :A3$=LEFT$(AT$,2)
620 C1=VAL(A3$):XT=INT(5*C1/R)
    :IF XT<10 THEN 640
630 XT=9
640 X$=STR$(X):X$=RIGHT$(X$,1)
    :R$=R$+X$:X=XT:A2$=AT$
650 NEXT I:CO=5:AT$=""
660 LX=(LX/2)+1:R$=LEFT$(R$,LX)
    :A1$=""
670 FOR I=LX TO 1 STEP -1
    :X1$=MID$(R$,I,1)
680 X=VAL(X1$)+CO:CO=0
    :IF X<10 THEN 700
```

```
690 X=X-10:CO=1
700 X1$=STR$(X):X1$=RIGHT$(X1$,1)
    :A1$=X1$+A1$
710 NEXT I:LR=INT((AL+1)/2)
    :LX=LX-1:IF CO=0 THEN 730
720 CO$=STR$(CO):CO$=RIGHT$(CO$,
    1):A1$=CO$+A1$:LR=LR+1
730 R$=LEFT$(A1$,LX)
    :IF LR>0 THEN 770
740 IF LR=0 THEN 760
750 F=ABS(LR):FOR I=1 TO F
    :R$=F$+R$:NEXT I
760 R$=F$+D$+R$:GOTO 810
770 RR=ABS(LX-LR):IF LX>LR THEN
    800
780 IF LX=LR THEN 810
790 FOR I=1 TO RR:R$=R$+F$:NEXT I
    :GOTO 810
800 RL$=LEFT$(R$,LR)
    :RR$=RIGHT$(R$,RR)
    :R$=RL$+D$+RR$
810 PRINT:PRINT R$:END
```


VIC-20 & 64

What *is* a Cheatsheet, anyway?

Leroy's Cheatsheet™ Keyboard overlays are durable plastic-coated templates. When simply laid on your VIC-20 and -64 keyboards, the **Leroy's Cheatsheet™** surrounds the keys with essential information, placing your most valuable programming tool at your fingertips.

At **Cheatsheet Products™** we take the time to learn and use each program before designing a keyboard overlay. Not only are our overlays designed using easy to follow instructions and illustrations, but all commands are available and many extras are added to make programming easy and fun.

Leroy's Cheatsheet™ overlays make it all easy for only **$3.95.**

Please send me the following Leroy's Cheatsheet™ keyboard overlays

20	64		20	64	
☐	☐	Programmer's Aid[1]	☐	☐	Graphic printer (1515 & 1525)[1]
☐		Vicmon[1]	☐		UMI Wordcraft 20
☐		Super Expander[1]	☐		HES Vic Forth
☐		Vic Typewriter[1]	☐	☐	HES Writer
☐		Victerm 1[1]		☐	Wordpro 3 plus
	☐	Term 64[1]		☐	Easy Script[1]
☐	☐	Quick Brown Fox	☐	☐	Basic
	*☐	Hesmon		*☐	Paper Clip
	*☐	Calc Result (advanced)		*☐	Script 64
	*☐	Pilot (Commodore)		☐ *☐	PractiCalc 64/plus

*NEW

Send check or money order plus $ 1.00 (postage and handling)
PA residents add 6% sales tax.

Name _____

Address _____

City _____ State _____ Zip _____

(1) Products of Commodore Business Machines, Inc. VIC-20 and CBM-64 are products of Commodore Business Machines, Inc.

CHEATSHEET PRODUCTS™
P.O. Box 8299 Pittsburgh PA. 15218 (412) 456-7420

TEACHERS

useful programs for COMMODORE 64™ and PET®

MASTER GRADES — The complete grading and attendance system. Designed for the novice computer user. Handles up to 200 students in one file and cumulative points to 9999. Even prints 3 kinds of progress notes to parents. Thousands now in use. Disk only.
Still only $39.50

TESTMASTER — The complete test and quiz development system. Produces tests with an assortment of up to 100 **true-false, multiple choice, completion,** and **short answer items.** End your test re-typing forever. Disk only. **$35.00**

FOOTBALL SCOUT — Get ready now for the coming season and beat your opposition. Accepts 15 user-defined categories and up to 50 plays per game. Six scouting files may be linked to give a report showing 300 plays on the team scouted. Disk only. **$79.50**

These programs require 32K in the Commodore PET.

Please write for our flyer of other useful and educational programs.

VISA and Mastercard accepted.

Please add $2.00 per disk for postage and handling.

MIDWEST SOFTWARE
- A DIVISION OF ZERO-ONE, LTD.
Box 214 • Farmington, MI 48024
Phone (313) 477-0897

TM AND ® ARE REGISTERED TRADEMARKS OF COMMODORE BUSINESS MACHINES, INC.

NEW Home Workstation!

Improve Your Commodore Home Computer System...

The C•A•T will organize your total computer system and provide a large work area with plenty of storage space for software and manuals.

- **COMPACT** Requires minimum floor space.
- **FUNCTIONAL** Centralizes hardware and software. Large work area. Right or left hand oriented. Printer paper feeds through shelf.
- **COMFORTABLE** Standard typing height, keyboard shelf. Adjustable leg levelers.
- **ATTRACTIVE** Designed to fit any decor. Vinyl, walnut finished shelves for maintenance free beauty.
- **STURDY** Engineered for years of service. 16 gauge steel. 5/8" laminated, flakeboard shelves. 56 lbs.
- **EASY TO ASSEMBLE** Assembles in minutes.

NOW AVAILABLE AT A SPECIAL INTRODUCTORY PRICE **$149.95**
Visa or Master Charge — Shipping charges collect.

COMMODORE APPROVED PRODUCT

Natl. 800-872-3333 Call Toll Free PA 800-292-9660

Name _____
Address _____
City _____ State _____ Zip _____
Phone _____ MC ☐ VISA ☐
Card No. _____ Expiration Date _____
Check enclosed, $ _____ for _____ Computer Access Tables, Model C
PA residents add 6% sales tax.

Order today from: Suckle Manufacturing Corp.
733 Davis Street Scranton, PA 18505

C•A•T

C*A*R*S

by new leaf inc.™
are your car costs
out of control?

We can help! Do you realize how much your car costs you? Do you know when your car needs a tune-up? Whether for an individual family or a fleet of vehicles, the cost of car ownership can be one of your biggest expenses.

It's not always easy to actually keep track of the real costs involved. Now you can with **C•A•R•S**! New Leaf's™ Car Analysis Reporting System was designed to determine the efficiency and cost of ownership of your vehicles.

While most other programs of this nature lump all the items together, we keep track of ownership and operating costs separately as well as together. You can now graph and monitor all of the costs involved.

C•A•R•S runs on both the Commodore 64™ and the VIC-20™ with 24K added. This means you can "step-up" from a VIC-20™ to a Commodore 64™ without needing to purchase a new program. Plus, as with our other programs, you even have the option of printed reports on your Commodore™ compatible printer.

You can now graphically see how effective any adjustments have been and our colorful bar graphs will let you see trouble spots at a glance.

C•A•R•S is an excellent tool for small businesses that need to monitor the cost of your fleet of vehicles. Using this program you can keep track of an unlimited number of cars and/or trucks.

Cassette Version . . . 35.50
Disk Version 39.50

Coming soon — The Magic Shoppee!

also available . . .
BIORHYTHM+ Cassette Only . . . 14.50
please state VIC-20™ or Commodore 64™

ESP•Calc (runs on both VIC-20™ and Commodore 64™)
Cassette Version . . 43.50 Disk Version . . 47.50

new leaf inc.

Product ordered _____
☐ VIC-20™ ☐ Commodore 64™
☐ Disk ☐ Cassette
Amount enclosed _____
Name _____
Address _____
City _____ State _____ Zip _____
Phone Number ( ) _____
MC or VISA # _____
Expiration date _____
Interbank # (MC) _____
Missouri residents add 5.125% sales tax
120 Lynnhaven • Belleville, IL 62223

Mastercard and VISA customers, please add 3%

Curve Fitting Simplified

by Bryan P. Herve

QUESTION: Can you name a method for analyzing data that is easy to perform on almost any computer, yet has applications in areas ranging from sales forecasting to designing complex machine parts?

Give yourself an A+ if you answered "curve fitting". A typical problem that can be solved by curve fitting occurs when the boss hands you the sales figures for the last ten years and says, "Here, study this data and make an estimate of next year's sales for the shareholders' report. I need it by this afternoon."

Curve fitting, more formally known as least-squares regression analysis, is a statistical technique useful for determining the mathematical relationships that may exist among a collection of data points. The simple objective is to reduce a large unmanageable volume of data to one convenient equation.

The method is most often applied to data that exhibits "scatter", meaning that one line cannot connect all of the points. This occurrence is shown in Figure 1. The least-squares technique attempts to find one line such that the distance from each point to the line is kept as small as possible. To avoid negative offsets (the point is below the line), which would cancel positive offsets, the square of each distance is used and the sum of the squares is minimized, hence the name "least-squares" analysis.

Straight lines are represented algebraically by Y=a+bx, where "a" is the Y intercept and "b" is the slope (remember all this from geometry?). For any set of data, the job of regression analysis is to find the values of "a" and "b" that yield the best fitting line.

Mathematically, the answer can be found as follows:

$$b = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sum X^2 - \frac{(\sum X)^2}{N}}$$

$$a = \frac{\sum Y}{N} - b\frac{\sum X}{N}$$

The "$\sum$" character is shorthand for "summation" and may be represented in BASIC (for $\sum$ X) as:

SX=O: FOR I=1 TO N : SX=SX+X(I): NEXT I

where "SX" will be the value of $\sum$ X when the loop is finished.

Other curve types may also be represented by modifications of the linear model. These include:
Exponential: Y=ae↑bx, which is LN(Y)=LN(a)+bX
Power: Y=ax↑b, which is LN(Y)=LN(a)+bLN(X)
Logarithmic: Y=a+bLN(X)

Listing 1 is a short BASIC 4.0 program written on a Commodore CBM 8032 that implements all four of these curve-fitting models. The data points shown in Figure 1 are included in the data statement (line 190) as a sample problem. The number calculated for variable R2 (see line 5300) is a statistical function called the "coefficient of determination". It represents the quality of the computed values for "a" and "b" in the regression equation. The value of R2 can vary between 0.0 and 1.0 with 0.0 being a poor fit and 1.0 a perfect fit.

Figure 2 shows the results of testing each available curve type on the given data set. The power model has the largest value of R2 and is the best choice to mathematically represent the data. The estimate for "next year's sales" is found by inserting the year value into the regression function. Using immediate mode on the CBM, type:

print 19.5429545*(11↑.71851532)

and press RETURN. The answer is 109.457826, or for the scale units used, $109,457,826.00 which should be rounded to $109,000,000.00.

Give curve fitting a try. It's not hard to use, but when properly applied it can save a great deal of work. **C**

Bryan P. Herve is a chemical engineer with 12 years of experience using small computers in technical applications.

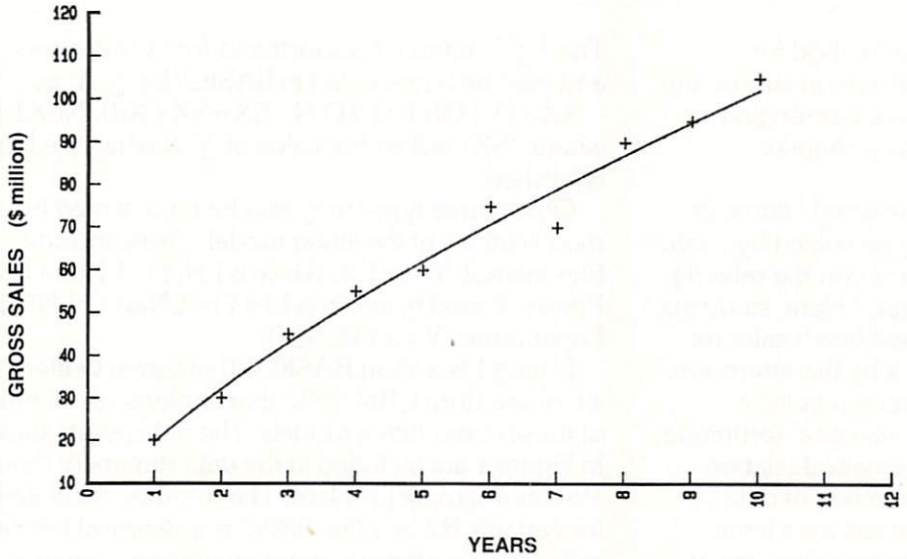


Figure 1: Sample Curve Fitting Problem

```
POINT #          X DATA          Y DATA
 1                1               20
 2                2               30
 3                3               45
 4                4               55
 5                5               60
 6                6               75
 7                7               70
 8                8               90
 9                9               95
 10               10              105

============================================================
Linear Model:    Y=a+bX

a=   14.3333333
b=   9.12121213

r2=   .9773887
```

Figure 2. Results of the sample problem shown in Figure 1

```
==========================================
Exponential Model:   Y=a*exp(bX)

a=   23.205381
b=   .166091969

r2=   .900721182

==========================================
Logarithmic Model:   Y=a+bLN(X)

a=   9.19646103
b=   36.6141607

r2=   .923177384

==========================================
Power Model:   Y=aX^b

a=   19.5429544
b=   .71851532

r2=   .988075121
```

Curve Fitting

```
10 REM LEAST-SQUARES REGRESSION
   ANALYSIS
20 REM
30 REM BY:BRYAN P. HERVE
40 REM SIERRA DIGITAL RESEARCH,
    INC.
50 REM 100 WASHINGTON STREET
60 REM    SUITE 104
70 REM RENO, NV  89503
80 REM
90 REM ------------------------
100 REM
110 REM   DEFINE THE NUMBER OF P
    OINTS & DIMENSION DATA ARRAYS
120 N=10
130 DIM X(N),Y(N),XC(N),YC(N)
140 REM   LOAD THE DATA POINTS I
    NTO THE ARRAYS
150 RESTORE:D$="================
    ============================"
160 FOR I=1 TO N
170 READ X(I),Y(I)
180 NEXT I
181 REM RESERVE LINES 182 TO 198
     FOR DATA STATEMENTS
190 DATA 1,20,2,30,3,45,4,55,5,
    60,6,75,7,70,8,90,9,95,10,105
199 REM
200 CLOSE 1:CLOSE 2:OPEN 2,4,7
    :PRINT#2:OPEN 1,4,0
    :REM COMMODORE 4022
205 REM  USE OPEN 1,
```

```
    3 IF YOU DON'T HAVE A PRINTE
    R & CHANGE LINE 260 TO REM
210 PRINT#1,"POINT #","X DATA",
    "Y DATA"
220 FOR I=1 TO N
230 PRINT#1,I,X(I),Y(I)
240 NEXT I
250 PRINT#1:PRINT#1
260 PRINT "[CLEAR]"
    :REM  CLEAR SCREEN FOR MENU
270 PRINT TAB(25);"[RVS]
    REGRESSION ANALYSIS PROGRAM
    [RVOFF]"
280 PRINT :PRINT "YOU HAVE ";N;
    " DATA POINTS ENTERED":PRINT
    :PRINT
290 PRINT TAB(25);"(1) LINEAR
    [SPACE5]: Y=A+B*X"
300 PRINT TAB(25);"(2) EXPONENTI
    AL: Y=A*E^(B*X)"
310 PRINT TAB(25);"(3) LOGARITHM
    IC: Y=A+B*LN(X)"
320 PRINT TAB(25);"(4) POWER
    [SPACE6]: Y=A*X^B"
330 PRINT
340 PRINT TAB(25);"(5) STOP"
    :PRINT:PRINT
350 PRINT TAB(25);"[RVS]
    SELECT A REGRESSION MODEL (1
     TO 4) OR PRESS 5 TO STOP
    [RVOFF]"
360 GET I$:IF I$="" THEN 360
370 K=ASC(I$):IF K=53 THEN 7000
    :REM STOP PROGRAM
380 IF K>48 AND K<53 THEN 400
390 GOTO 360
400 K=K-48:ON K GOTO 440,520,570,
    620 :REM K=CURVE TYPE SELECT
    ION
410 REM
420 REM LINEAR MODEL ********
430 REM
440 PRINT#1:PRINT#1,D$
    :PRINT#1,"LINEAR MODEL
    :[SPACE3]Y=A+B*X":PRINT#1
450 FX=0:FY=0:REM THESE FLAGS AR
    E USED TO INDICATE IF LOGARI
    THMS ARE
460 REM  NEEDED BY THE REGRESSIO
    N MODEL.
470 GOSUB 5000:REM PERFORM THE S
    TATISTICAL SUMS TO FIND A & B
480 GOTO 260
490 REM
500 REM EXPONENTIAL MODEL *****
510 REM
520 PRINT#1:PRINT#1,D$
    :PRINT#1,"EXPONENTIAL MODEL
    :[SPACE3]Y=A*EXP(B*X)"
    :PRINT#1
530 FX=0:FY=1:GOSUB 5000:GOTO 260
540 REM
550 REM LOGARITHMIC MODEL *****
560 REM
570 PRINT#1:PRINT#1,D$
    :PRINT#1,"LOGARITHMIC MODEL
    :[SPACE3]Y=A+B*LN(X)":PRINT#1
580 FX=1:FY=0:GOSUB 5000:GOTO 260
590 REM
600 REM POWER MODEL ******
610 REM
620 PRINT#1:PRINT#1,D$
    :PRINT#1,"POWER MODEL
    :[SPACE3]Y=A*X^B":PRINT#1
630 FX=1:FY=1:GOSUB 5000:GOTO 260
640 REM
4990 REM THIS SUBROUTINE CALCULA
     TES THE REGRESSION SUMS & C
     OEFFICIENTS
5000 TX=0:TY=0:REM THESE ARE FLA
     GS TO INDICATE 0 OR NEGATIV
     E VALUES
5010 FOR I=1 TO N
5020 XC(I)=X(I):YC(I)=Y(I)
     :REM COPY THE ORIGINAL DATA
5030 IF XC(I)<=0 THEN TX=1
5040 IF YC(I)<=0 THEN TY=1
5050 NEXT I
```

```
5060 REM XC() AND YC() ARE USED
     TO MAKE COPIES OF THE DATA
     ARRAYS
5070 IF FX=0 THEN 5120
     :REM X LOG VALUES NOT NEEDED
5080 IF TX=1 THEN 6000
     :CAN'T TAKE LOG OF 0 OR NEG
     ATIVE NUMBERS
5090 FOR I=1 TO N
5100 XC(I)=LOG(X(I))
5110 NEXT I
5120 IF FY=0 THEN 5170
     :REM Y LOG VALUES NOT NEEDED
5130 IF TY=1 THEN 6000
     :REM CAN'T TAKE LOG OF 0 OR
      NEGATIVE NUMBERS
5140 FOR I=1 TO N
5150 YC(I)=LOG(Y(I))
5160 NEXT I
5170 REM BEGIN CALCULATING THE S
     UMMATION QUANTITIES
5180 SX=0:SY=0:XY=0:X2=0:Y2=0
5190 FOR I=1 TO N
5200 SX=SX+XC(I):REM SUM OF THE
     X VALUES
5210 SY=SY+YC(I):REM SUM OF THE
     Y VALUES
5220 XY=XY+XC(I)*YC(I)
     :REM SUM OF THE XY VALUES
5230 X2=X2+XC(I)^2:REM SUM OF TH
     E X VALUES SQUARED
5240 Y2=Y2+YC(I)^2:REM SUM OF TH
     E Y VALUES SQUARED
5250 NEXT I
5260 B=(XY-(SX*SY)/N)/(X2-(SX^2)
     /N):REM THE 'B' REGRESSION
     VALUE
5270 IF K=1 OR K=3 THEN A=(SY-B*
     SX)/N:REM THE 'A' REGRESSIO
     N VALUE
5280 IF K=2 OR K=4 THEN A=EXP((S
     Y-B*SX)/N)
5290 REM  CALCULATE THE COEFFICI
     ENT OF DETERMINATION (R^2)
5300 R2=((XY-(SX*SY)/N)^2)/((X2-
     (SX^2)/N)*(Y2-(SY^2)/N))
5310 PRINT#1,"A= ";A
5320 PRINT#1,"B= ";B:PRINT#1
5330 PRINT#1,"R2= ";R2:PRINT#1
5340 RETURN
5990 REM ERROR TRAP FOR 0 OR NEG
     ATIVE DATA IN LOGRARITHMS
6000 PRINT#1,"THIS MODEL CANNOT
     BE EVALUATED DUE TO:"
6010 IF TX=1 AND FX=1 THEN PRINT
     #1,"[SPACE20]0 OR NEGATIVE
     X VALUES"
6020 IF TY=1 AND FY=1 THEN PRINT
     #1,"[SPACE20]0 OR NEGATIVE
     Y VALUES"
6030 PRINT#1:PRINT CHR$(7)
     :GOTO 260
6980 REM THIS SEGMENT DISPLAYS T
     HE DATA STATEMENT BLOCK FOR
      EDITING
6990 REM      AND STOPS THE PROG
     RAM
7000 CLOSE 1:CLOSE 2
     :PRINT "[CLEAR]"
     :PRINT "ALL DONE!":PRINT
     :PRINT
7010 PRINT "IF YOU NEED TO EDIT
     YOUR DATA, MOVE THE CURSOR
     TO THE DESIRED"
7020 PRINT "LOCATION AND MAKE TH
     E CHANGES.[SPACE2,SHFT P]
     RESS RETURN TO ENTER THE NE
     W"
7030 PRINT "DATA STATEMENT[S]
      INTO THE PROGRAM MEMORY.
     [SPACE2]THEN TYPE 'RUN' "
7040 PRINT "AND PRESS RETURN
     [SPACE2](BE SURE THE PRINTE
     R IS ON!)."
7050 PRINT:PRINT:LIST 182-198
     :PRINT CHR$(7):PRINT:STOP
```

Smooth Scrolling Right-to-Left

by Brian Schott

An electronic marquee is used by many commercial enterprises to promote their products and services, scrolling messages across the video medium from right to left. You can create an electronic marquee using the feature on the Commodore 64 called "smooth scrolling".

Beginning on page 128 of the *Commodore 64 Programmer's Reference Guide* you'll find an explanation of smooth scrolling. The example listed on page 130 of the *Reference Guide* provides a good example of scrolling from the bottom of the screen. But the smooth scrolling provided automatically by the Commodore 64 hardware scrolls only eight bits at a time—a total of one screen character's distance. After the screen is smoothly scrolled eight bits, something else must shift the screen over into the next eight-bit position so that the next smooth scroll can be performed. To accomplish this, the example on page 130 of the *Reference Guide* uses the Commodore 64's PRINT command to shift the entire screen upward one line (remember that when a PRINT command occurs when the cursor is on the bottom line of the video screen, then the screen automatically rolls up one line).

But when you want to scroll from right to left, there is no BASIC command designed to shift the entire screen left one whole character. Therefore, I created the machine language (ML) program at the end of this article to perform the necessary shift quickly enough to make an electronic marquee on the Commodore 64.

The ML program contains two main parts. One part moves each character on the screen exactly one character position to the left. But this part is not enough since the first screen position on each screen line is stored in memory just to the left of the last screen position of the line which appears just above it. So another procedure is required to keep the marquee message from creeping onto the lines above after it fills its first line.

The second part of the ML program takes care of this by forcing a space character into the (offscreen) lefthandmost position of the screen. The space character is invisible when it appears on the next screen line. The padding of a space character is performed first. Then each character is shifted to the left one position.

To make the electronic marquee more easily visible I later added a feature that creates large letters. You'll find that program on page 93-94.

I developed these ML programs using the Eastern House Software program MAE. To use either one, first load and run the ML (data) programs. Then load and run the BASIC. C

Scroll Left Data

```
800 FOR ADRES=49152 TO 49237
    :READ DATTA:POKE ADRES,DATTA
    :NEXT ADRES
49152 DATA 160, 0, 169, 0, 133,
      251
49158 DATA 169, 4, 133, 252, 162,
      24
49164 DATA 24, 165, 251, 105, 40,
      133
49170 DATA 251, 165, 252, 105, 0,
      133
49176 DATA 252, 169, 32, 145,
      251, 202
49182 DATA 16, 236, 160, 1, 162, 3
```

```
49188 DATA 169, 0, 133, 251, 133,
      253
49194 DATA 169, 4, 133, 252, 133,
      254
49200 DATA 165, 253, 208, 2, 198,
      254
49206 DATA 198, 253, 177, 251,
      145, 253
49212 DATA 200, 208, 249, 230,
      252, 230
49218 DATA 254, 177, 251, 145,
      253, 200
49224 DATA 202, 208, 237, 177,
      251, 145
49230 DATA 253, 200, 192, 232,
      208, 247
49236 DATA 96, 145, 253, 200,
      192, 232
63000 END
```

Scroll Left BASIC Program

```
2 REM THIS PROGRAM ASSUMES THAT
  SCRLEFT DATA IS IN PLACE
4 IF PEEK(12*4096+5)<>251 THEN P
  RINT"YOU MUST RUN SCRLEFT DATA
   FIRST!!":STOP
10 POKE 53270,PEEK(53270)AND 247
20 PRINT CHR$(147)
30 FOR X=1 TO 39:PRINT CHR$(32);
   :NEXT
40 POKE 53270,(PEEK(53270)AND 24
   8)+7
45 K=K+1
50 SYS 12*4096:PRINT MID$( "HELL
   O ",K,1)CHR$(157);
55 IF K=6 THEN K=0
60 FOR P=6 TO 0 STEP-1
70 POKE 53270,(PEEK(53270)AND 24
   8)+P
80 FOR X=1 TO 50:NEXT
90 NEXT:GOTO 40
```

Marquee Data

```
800 FOR ADRES=49152 TO 49495
    :READ DATTA:POKE ADRES,DATTA
    :NEXT ADRES
49152 DATA 127, 127, 127, 127,
      127, 127
49158 DATA 127, 127, 160, 0, 169,
      0
49164 DATA 133, 251, 169, 4, 133,
      252
49170 DATA 162, 24, 24, 165, 251,
      105
49176 DATA 40, 133, 251, 165,
      252, 105
49182 DATA 0, 133, 252, 169, 32,
      145
49188 DATA 251, 202, 16, 236,
      160, 1
49194 DATA 162, 3, 169, 0, 133,
      251
49200 DATA 133, 253, 169, 4, 133,
      252
49206 DATA 133, 254, 165, 253,
      208, 2
49212 DATA 198, 254, 198, 253,
      177, 251
```

```
49218 DATA 145, 253, 200, 208,
      249, 230
49224 DATA 252, 230, 254, 177,
      251, 145
49230 DATA 253, 200, 202, 208,
      237, 177
49236 DATA 251, 145, 253, 200,
      192, 232
49242 DATA 208, 247, 96, 0, 31, 3
49248 DATA 247, 96, 120, 0, 60,
      102
49254 DATA 102, 102, 102, 102,
      60, 0
49260 DATA 169, 0, 141, 99, 192,
      14
49266 DATA 98, 192, 46, 99, 192,
      14
49272 DATA 98, 192, 46, 99, 192,
      14
49278 DATA 98, 192, 46, 99, 192,
      169
49284 DATA 208, 72, 169, 0, 72,
      173
49290 DATA 99, 192, 72, 173, 98,
      192
49296 DATA 72, 32, 50, 193, 133,
      252
49302 DATA 132, 251, 169, 100,
      133, 253
49308 DATA 169, 192, 133, 254,
      120, 169
49314 DATA 51, 133, 1, 160, 7,
      177
49320 DATA 251, 145, 253, 136,
      16, 249
49326 DATA 169, 55, 133, 1, 88,
      24
49332 DATA 173, 22, 208, 41, 248,
      105
49338 DATA 7, 141, 22, 208, 160,
      7
49344 DATA 162, 7, 177, 253, 10,
      62
49350 DATA 0, 192, 202, 16, 249,
      136
49356 DATA 16, 242, 162, 7, 24,
      173
49362 DATA 94, 192, 105, 0, 133,
      251
49368 DATA 173, 95, 192, 105, 4,
      133
49374 DATA 252, 160, 7, 152, 72,
      160
49380 DATA 0, 169, 32, 62, 0, 192
49386 DATA 144, 2, 169, 81, 145,
      251
49392 DATA 56, 165, 251, 233, 40,
      133
49398 DATA 251, 165, 252, 233, 0,
      133
49404 DATA 252, 104, 168, 136,
      16, 223
49410 DATA 138, 72, 162, 6, 24,
      173
49416 DATA 22, 208, 41, 248, 134,
      253
49422 DATA 101, 253, 141, 22,
      208, 202
49428 DATA 72, 138, 174, 93, 192,
      160
49434 DATA 0, 136, 208, 253, 202,
      208
49440 DATA 248, 170, 104, 16,
      231, 32
49446 DATA 8, 192, 104, 170, 202,
      16
49452 DATA 163, 96, 147, 192,
      120, 0
49458 DATA 104, 141, 46, 193,
      104, 141
49464 DATA 47, 193, 104, 141, 48,
      193
49470 DATA 104, 141, 49, 193,
      104, 24
49476 DATA 109, 48, 193, 168,
      104, 109
```

```
49482 DATA 49, 193, 170, 173, 47,
      193
49488 DATA 72, 173, 46, 193, 72,
      138
49494 DATA 96, 96, 104, 170, 202,
      16
63000 END
```

Marquee BASIC Program

```
2 REM THIS PROGRAM ASSUMES THAT
  MARQUEE DATA IS IN PLACE
4 IF PEEK(12*4096+12)<>133 THEN
  PRINT"YOU MUST RUN MARQUEE DAT
  A FIRST!!":STOP
5 B=12*4096
8 CHAR=B+ 98
15 PRINT CHR$(142);
   :REM SWITCH TO UPPERCASE
20 PRINT"[WHITE]":POKE 53281,1
   :PRINT"[CLEAR]":POKE 53281,0
25 POKE B+ 93,70:REM CONTROL SPE
   ED: 1,2,3,4,...,255,
   0  FAST TO SLOW
30 A$="THIS IS BRIAN'S MESSAGE
   :[SPACE2]LOVE YOUR NEIGHBOR.
   [SPACE2]"
35 MSG=LEN(" "+A$+"[SPACE2]")
40 K=K+1
48 TMP =ASC(MID$( " "+A$+"
   [SPACE2]",K,1)):IF TMP>  64 T
   HEN TMP=TMP-64
50 POKE CHAR,TMP:SYS B+108
   :PRINT CHR$(157);
60 IF K=MSG THEN K=0
80 GOTO 40
```

Reduction of an actual sign

The Banner Machine™

For the **Commodore 64** (4 extra fonts available). For the **VIC-20** with 24K memory (2 extra fonts available). • Use on any Gemini or Epson MX with Graftrax or the FX and RX printers. Also Commodore 1525E and Banana with the C-64. • Menu-driven program operates like a word processor. • Makes signs up to 13" tall by any length. • Makes borders of widths up to ¾". • 8 sizes of letters from ¾" to 8" high. • Proportional spacing; Automatic centering; Right and left justifying. • $49.95 Tape or Disk (Specify computer equipment)

For the Commodore 64:

Chessmate 64 Learning tool to analyze master and book games and openings. Create chess movies. Memorize and recall board positions. Save and print games. $29.95

Grade Organizer Teachers—store grades for 6 classes, up to 40 students each, 680 grades per student. Print interim and final reports, class rosters, and more! Disk $39.95

CTRL-64 Permits listing of C-64 programs on non-Commodore printers. Lists control symbols in readable form. Tape or disk $24.95

Space Raider An amazing arcade simulation. Your mission is to destroy the enemy ships. $19.95

Microbroker Exciting, realistic and educational stock market simulation. $34.95 Tape or Disk

Preschool Educational Programs ABC Fun and 123 Fun have bright color and action. **Each** $14.95

Formulator A scientific calculator for repetitive arithmetic computations. Save formulas. Ideal for chemistry, physics, or engineering students. $39.95

MUPOS Create a file of up to 9 basic programs from keyboard/tape/disk and run each program selectively. Combine programs. Renumber BASIC lines. Note pad, and more. Tape or disk $29.95

Screendump Print a copy of the C-64 screen by pressing two keys. Machine-language program is compatible with most software. $19.95

Toll Free for Orders Only: 800-555-1212
HOURS: 10 a.m. to 4 p.m. Mon.—Sat.

Cardinal Software™

Virginia Micro Systems
13646 Jeff Davis Hwy.
Woodbridge, VA 22191
Phone (703) 491-6502
Ask for free catalog

Commodore 64 and VIC-20 are registered trademarks of Commodore Electronics Ltd.

Standard Screen Plotting
on the Commodore 64

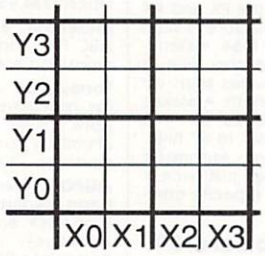
by Rolf L. Miller

Plotting formula graphs such as sine waves, exponential curves and the like is usually accomplished in the bit-mapped mode. However, when the high resolution of bit mapping is not needed, being able to plot without leaving the text mode is handy.

Standard screen plotting can be accomplished with little effort by using the complete quarter-space character set. The key to the task is in dealing with the screen as if it were an 80-by-50 plotting grid so that each quarter space can be properly referenced. Since standard screen memory allows only full-byte addressing, the grid must be simulated within the routine.

The screen, then, is viewed as if it contained 80 columns, 0 to 79, left to right, and 50 rows, 0 to 49, bottom to top. These provide the X and Y coordinates respectively necessary for plotting.

Each character space then has four references. For example, the bottom left hand space will be viewed as X0 Y0, X0 Y1, X1 Y0, X1 Y1:



The four quadrants of each space are assigned labels 0 to 3 as shown:

2	3
0	1

The order has to do with the following order of the 16 possible combinations of the quarter space characters.

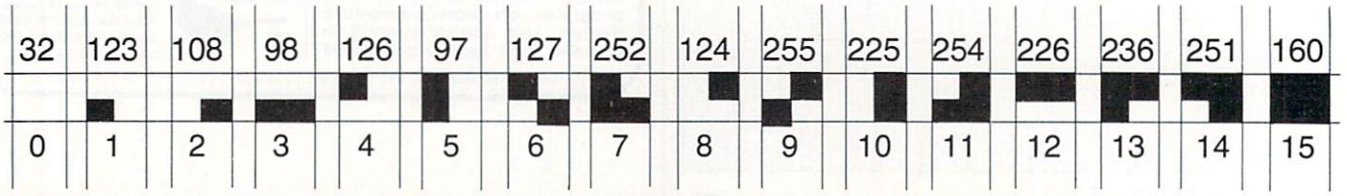
The numbers above the characters are their screen codes. The numbers under the characters are their element references as they will appear in a numerical array. They will be stored in the array by their screen code values as follows:

```
10 PRINT"[CLEAR]":DIM PC(15)
   :FOR E=0 TO 15:READ C:PC(E)=C
   :NEXT
20 DATA 32,123,108,98,126,97,127,
   252,124,255,225,254,226,236,
   251,160
```

The character in quotes is shift CLR/HOME to clear the screen for the demonstration routines to come.

Now, here is the trick. 2↑0 is one, 2↑1 is two, 2↑2 is four, and 2↑3 is eight. During plotting, the resulting value of two to the power of the quadrant's label being pointed to is used to determine which element of the array—and, thus, which character—is applicable. Viewing a space with the possible power values noted therein, it will be seen that the corresponding array element holds the proper character value. POKing that value to the space's screen memory location will display the character.

4	8
1	2



For instance, if quadrant three is being pointed to—as would be the case if, say, the coordinates were X1 Y1—2↑3 will yield eight. Element eight of the array holds value 124, which will produce the proper character when POKEd to the screen:

Of course, during plotting, more than one quadrant of a particular space may eventually be pointed to. When this occurs, the resulting character POKEd must represent both that which was already there plus the one being added. For example, if the space which already contains the character of element eight then has quadrant zero pointed to, the result must be the character of element nine.

This is simply accomplished by adding one (2↑0) to the eight already represented. Looking at the value each quadrant takes (1, 2, 4, 8), it is readily seen that every possible combination of characters represented in the array's elements (one through 15) is available by adding the values together as determined during plotting.

The plotting routine, therefore, takes the X and Y coordinates generated by the particular formula being used, calculates the screen memory address of the space in which the coordinates reside, and determines which quadrant of the space is being pointed to. The screen address is then PEEKed and the array element represented therein discovered. Following that, it is determined whether or not the quadrant being pointed to has been filled. If so, the calculation and POKing of the new character representation is skipped. It is reminded that the X and Y coordinates are column and row designations respectively.

```
100 SA=(24-INT(Y/2))*40+1024+INT
    (X/2)
110 X=X-(INT(X/2)*2)
    :Y=Y-(INT(Y/2)*2)
120 Q=0:IF X AND Y THEN Q=3
130 IF X AND Y=0 THEN Q=1
140 IF Y AND X=0 THEN Q=2
150 PE=PEEK(SA):FOR E=0 TO 15
    :IF PE=PC(E)THEN N=E
160 NEXT
170 F=INT(N/2^Q):IF F-(INT(F/2)*
    2)<>0 THEN 190
180 N=N+2^Q:POKE SA,PC(N)
    :POKE SA+54272,1
190 NEXT:IF S=1 THEN S=0:GOTO 50
200 INPUT Q$:RUN
```

The second POKE in line 180 POKEs the color (one—for white—in this case) to color memory. The test in line 190 following the NEXT statement is necessary only for this particular demonstration. The line 200 INPUT statement allows rerunning by just pressing RETURN.

The lines 30 through 90 following are only for illustration. The actual formula one might use will, of course, be dictated by the task at hand.

```
30 INPUT"SELECT: (1) (2) (3) (4)
   (5)";S:PRINT"[CLEAR]"
   :ON S GOTO 40,60,70,80,90
40 FOR P=0 TO 35:X=P:Y=35-P
   :GOTO 100
50 FOR P=0 TO 50:Y=P:X=0+P
```

(continued on page 100)

Indexer

by W. J. Crowley

INDEXER is a simple, easy to use program that will help you organize many kinds of information and retrieve it using a variety of search keys. The program can easily be tailored to keep an inventory of your personal valuables, a computer program library, an index of merchants or suppliers you deal with regularly or many other kinds of text related facts.

The following listing illustrates how I organized a list of club officials for easy access. You don't usually need this kind of data on-line for immediate access, but it is nice to be able to search or update it easily. This example includes a lot of REM statements for easy understanding but you could omit them if you wish.

Since I use several versions of INDEXER for different things, I've labeled each version with the filename at line ten. The main program begins at line ten and consists of only three subroutines, with an END at line 80. All DATA statements follow immediately, from line 1000 through line 19999.

The initialization routine at line 20000 collects up all the variables for page headers, screen size, record length, and so forth. You can tailor the program for most applications by making changes here, leaving the remaining lines intact. Commodore BASIC doesn't require the DIM statement at line 20020, but I always include it for transportability of code.

The search-key routine at line 20200 is fairly self explanatory. Line 20230 handles a carriage return answer to the "key" question, that would otherwise produce unpredictable or unwanted results. With this test, a carriage return response simply returns control to the main program.

The search-and-display routine at line 20400 does the real work of the program, and deserves a closer look. Line by line, here's what it does:

20410 Traps carriage returns, same as line 20230.
20420 Reads data, and returns to BASIC if at the end of the list.
20430 Determines if the data record (string) contains the search-key in the correct columns. Returns to the READ statement if the key isn't found.
20440 Prints data records that contain the search-key, adds one to the number of lines printed, and repeats the READ statement if screen display space is available.
20450 Prints an appropriate message if the screen is full.
20460 Waits for a key-press to proceed.
20470 After a key-press, resets the screen line counter and returns to the READ statement.
20480 When the end of the data list is reached, prints a message if no correct search-keys were found.
20490 Returns to the main program.

Now you should type in the program and test it. When testing, I suggest you try the carriage return only answer; INDEXER should end and return you gracefully to BASIC. Search for something that's not in the DATA list, like the number 1296 starting in column 26. You should get the "no matching keys" message, and return to BASIC. A search for an apostrophe in column 37 should produce a list of all data items, since all match this key. Try searching for an asterisk in column 19. A list of all committee chairmen should result.

At this point, the "no frills" program is complete. Those of you with printers and disk drives may want to modify INDEXER to print hardcopy reports (as I do, at work) or to read the data list from disk instead of carrying it in each program.

If some search-keys are used often, you may want to program the function keys to select the starting column number, instead of typing the number for each search. For data bases that you will be searching repeatedly, you may want to add an instruction to restart the program automatically, instead of having to type "RUN" for each inquiry. A

few minutes of exploring and experimenting will usually tailor an INDEXER to meet your needs for a simple but effective information retrieval and management system. Besides, making a useful modification is fun. C

Indexer

```
10 REM *** SAVED @ INDEXER ***
20 REM       W J CROWLEY
30 REM **** MAIN PROGRAM *****
40 REM
50 GOSUB 20000:REM - INITIALIZE
60 GOSUB 20200:REM - GET SEARCH
   KEY
70 GOSUB 20400:REM - SEARCH/DISP
   LAY
80 END:REM *** END OF PROGRAM ***
1000 REM ** DATA FILE STARTS HER
     E **
1010 DATA PRESIDENT            A S
     MYTHE          '84
1020 DATA VICE-PRESIDENT       E M
     ONTAINE        '86
1030 DATA SECRETARY            H M
     ORGAN          '84
1040 DATA TREASURER            G B
     OGGS           '85
1050 DATA OFFICE              *M M
     CBRIDE         '85
1060 DATA OFFICE               G B
     OGGS           '85
1070 DATA SWIMMING POOL       *W B
     LACKBURN       '85
1080 DATA SWIMMING POOL        C B
     ROWN           '86
1090 DATA MEMBERSHIP          *M F
     ABER           '86
1100 DATA MEMBERSHIP           E M
     ONTAINE        '86
1110 DATA GROUNDS/BUILDINGS   *D D
     ENNIS          '84
1120 DATA GROUNDS/BUILDINGS    W B
     LACKBURN       '85
1130 DATA GROUNDS/BUILDINGS    M F
     ABER           '86
1140 DATA TENNIS COURTS       *C G
     LITZKE         '86
1150 DATA TENNIS COURTS        W B
     ROWN           '85
1160 DATA DUES                *J S
     WANSON         '86
1170 DATA DUES                 H M
     ORGAN          '84
1180 DATA DUES                 M M
     CBRIDE         '85
1190 DATA PUBLIC RELATIONS    *C B
     ROWN           '86
1200 DATA PUBLIC RELATIONS     C G
     LITZKE         '86
1210 DATA SPECIAL EVENTS      *W B
     ROWN           '85
1220 DATA SPECIAL EVENTS       D D
     ENNIS          '84
19999 DATA END
20000 REM *** INITIALIZE ***
20010 R=40:REM - RECORD LENGTH
20020 DIM D$(R),H1$(R),H2$(R),
      H3$(R),KEY$(R)
20030 CT=0:PAGE=20:REM - SET LIN
      E COUNT       & PAGE SIZE
20100 H1$="[SPACE15]
```

(continued on page 100)

Indexer

(Continued from page 99)

```
      BOARD MEMBERS"
20110 H2$="[SPACE5]COMITTEE
      [SPACE12]MEMBER[SPACE5]RET"
20120 H3$="1[SHFT D] 20[SHFT D]
       37[SHFT D]"
20130 RETURN
20200 REM *** GET SEARCH KEY ***
20210 PRINT"[CLEAR]":PRINT
      :PRINT H1$:PRINT H2$,H3$
20220 PRINT:PRINT "SEARCH FOR WH
      AT KEY...":INPUT"[SPACE3]
      (WORD,LETTER,PHRASE)";KEY$
20230 IF LEN(KEY$)=0 THEN 20260
20240 INPUT "STARTING IN WHAT CO
      LUMN";COL
20250 PRINT "[CLEAR]":PRINT
      :PRINT H1$:PRINT H2$,H3$
20260 RETURN
20400 REM *** SEARCH/DISPLAY ***
20410 IF LEN(KEY$)=0 GOTO 20490
20420 READ D$:IF D$="END" GOTO 2
      0480
20430 IF MID$(D$,COL,
      LEN(KEY$))<>KEY$ GOTO 20420
20440 PRINT D$:CT=CT+1
      :IF CT<PAGE-2 GOTO 20420
20450 PRINT "[SPACE3]
      *** PRESS ANY KEY TO CONTI
      NUE ***"
20460 GET K$:IF K$="" GOTO 20460
20470 CT=0:GOTO 20420
20480 IF CT<1 THEN PRINT "
      [SPACE7]NO MATCHING KEYS F
      OUND."
20490 RETURN
```

Standard Screen Plotting

(Continued from page 97)

```
   :GOTO 100
60 FOR P=0 TO 60 STEP.5
   :A=P/30*[PI]:X=INT(25.5+24*SI
   N(A)):Y=INT(24.5+18*COS(A))
   :GOTO 100
70 FOR P=0 TO 109 STEP.5
   :A=P/30*[PI]:X=INT(EXP(P/25.5
   )):Y=INT(24.5+18*COS(A))
   :GOTO 100
80 FOR P=0 TO 159 STEP.5
   :A=P/30*[PI]:X=INT(P/2)
   :Y=INT(24.5+18*COS(A))
   :GOTO 100
90 FOR P=0 TO 79:X=P
   :Y=INT(20*SQR(P/16)):GOTO 100
```

The demonstration may now be run. Selecting number one will give a simple X,Y axis graph. When the plotting is complete, press RETURN to rerun. Number two will plot a circle—and herein will be found a bug in standard screen plotting.

The 25.5 and 24.5 represent the center X,Y coordinates chosen for this plot (line 60). It will then be noticed that 24 is added to the X value, but only 18 is added to Y. This offset is necessary to adjust the radius to the fact that characters are not square. That is, characters are longer than they are wide so that 40 spaces across by 25 lines down can be displayed on a square screen. Without an adjustment to this fact, the resulting plot, though mathematically correct, would not appear as a circle.

Finally, it is noted that removing the STEP .5 from those loops possessing it will speed up the plotting, but gives a coarser graph. **C**

Multi-File ASCII-to-WordPro Conversion

by Elizabeth Deal

WordPro™, in addition to its normal function of being a state-of-the-art word processing system, can be used as an excellent file editor. WordPro can write sequential files if the user asks for it. But it is not easy to edit such files directly. The difficulty is in bringing them back into the WordPro system, since this involves tricky maneuvers with variable blocks. We can avoid this by writing a conversion routine instead.

Similar conversion routines have been published before—Paul Higginbottom and Neil Harris had their versions in this magazine, for instance. The program presented here has two additional twists: (1) it handles long files by writing several WordPro files for each ASCII file, so that subsequent global processing can be done and (2) it does the ASCII-to-screen code conversion right before your eyes, without any conversion arrays or calculations. This method can be generalized to applications other than file conversion.

The most common candidates for a routine such as this are normal, sequential files and BASIC program listings (no graphics). The sequential files must be written with up to 255 characters separated by carriage returns. BASIC programs must be listed by the usual OPEN 3,8,3, "FILE, 5,W": CMD3:LIST ...PRINT#3:CLOSE3. Some computers put out a linefeed character in the first position of each line when listing a program, but this is ignored. Some programs (including WordPro) conveniently put out a quote as the initial character of each hunk of data. This also is ignored, but if you need to retain the quote, delete the IA=QT test in line 360.

In order to stay general—to be able to input any kind of data, including commas and colons—I have borrowed Jim Butterfield's "String Thing", which is a "better INPUT#" routine (see *The Transactor*, Volume 4). "String Thing" is particularly valuable where program listings are involved, otherwise INPUTS# would be unable to process full lines of BASIC text.

The output of the program consists of one or more WordPro files with the following naming convention: the initial "w." followed by a two-digit number, followed by the first ten characters of the input file name you specify. The number of files written depends on your WordPro system. My very old WordPro3 is good for 337 lines. Newer systems use less. You specify the file size you need.

Since the purpose of this exercise is file editing, I recommend that you ask for about 60% of the capacity. Each WordPro file gets a comment line as the first line of text. It contains the file name, so you can use it in saving via the backslash. Each file, except the last, ends with the link command to the next file, so you can work all the global commands.

Once you have edited the file, you can, of course, write it out again as a tight sequential file by specifying sequential output, and you can then scratch the WordPro files we just created, since they use lots of room and are no longer needed.

Program Description

Lines 140-180: Butterfield "String Thing". A$ must be a first defined variable. It must be 255 characters long so type line 140 exactly. Type DATA lines correctly. An error will cause a STOP in line 150. Fix and re-RUN if that happens.

Line 190: There must be at least 40 spaces within the quotes, so that the resulting BL$ is at least 80 characters long. This is vital because we must make all WordPro lines equal. The line length is always adjusted to a full line. The length depends on the system—40- or 80-column.

Line 200: Sets up the comment, next-file, and backarrow strings.

Lines 210-220: Create a phony load address, assign quote and linefeed characters to variables and check in which system the routine is running (40 or 80 columns). I have only tested this routine in my

40-column upgrade (BASIC 2/3) PET. You may have to fiddle with the code if it does not work in a BASIC 4 PET or the 80-column machines.

Lines 230-290: Ask for user input and verify the line count, initialize the drive(s) and open the read-file. MX is the variable that holds the maximum number of lines your WordPro can handle. You may use one or two disk drives; two permit larger files, of course. The file name may be input with an asterisk (*) at the end. The program ignores it.

Lines 300-320: Set up the first file name N$ and open that file for writing. The initial CLOSE is vital here, as we loop back to line 320 after a link to the next file is established in line 380.

Lines 320-380: File loop. Each new file consists of up to MX lines. All WordPro lines or sequential file records are handled in a line loop in lines 340-370 where we String-Thing-INPUT a record, fix it up, and use it. You may push "Q" to quit at any time. The linefeed/quote test is in line 360.

Lines 390-420: Count output lines and characters per line, print a string on the screen (what was input, comment or next-file directive), print the back arrow and write the disk by PEEKing the screen. This is our quick and dirty ASCII-to-WordPro (or screen code) conversion. Line 420 is very important, it makes all WordPro lines the same length which is the key purpose of this whole routine.

Lines 430-440: Disk error checking. We kill the whole project if there are disk errors. This could use improvements if it worries you. You could check for a "disk full" condition—that's probably the most important one to take care of. Whatever you do, make sure that if the program aborts for any reason a file may not be closed. Do not scratch it, validate (collect) the disk instead.

Line 450: Creates a file name, as described above. You may wish to change my "w." convention to whatever system you have created. I prefix all WordPro files with a "w." to keep me out of trouble.

Miscellaneous

The good old Canadian publication, *The Transactor* has been running an excellent WordPro column by Donna Green. The articles expand on the WordPro manual, showing in detail how to use various WordPro features. While all the features are, in some form, described in the manual itself, they are often cryptic. Donna's column elaborates on the uses and shows handy tricks. If you use WordPro for anything larger than a letter to your aunt, you should consider acquiring *The Transactor,* if not all issues, then at least the current Volume 4, of which issue five is a goldmine of invaluable charts and maps of all Commodore computers! Their address is: 500 Steeles Ave., Milton, Ontario, L9T 3P7, Canada. Phone: (416) 876-4741. C

ASCII-to-WordPro

```
100 REM"S=SAVE"ASC->WORDPRO",8
110 REM-----------
120 REM PET2/4 ASCII-SEQUENTIAL
    TO WORDPRO CONVERSION -> ELI
    ZABETH DEAL
130 REM-----------
140 A$="STRINGTHING JIM B"
    :A$=A$+A$+A$+A$+A$
    :A$=A$+A$+A$
150 FOR J=896 TO 933:READ X
    :POKE J,X:T=T+X:NEXT J
    :IF T<>5517 THEN STOP:***
160 DATA 160,2,177,42,153,134,0,
    200,192,6,208,246,162,1,32,
    198,255
170 DATA 32,228,255,201,13,240,
    11,164,139,145,137,200,132,
    139,196,136,208,238
```

```
180 DATA 76,204,255
190 BL$="[SPACE40]":BL$=BL$+BL$
200 CM$=CHR$(186)+"CM:"
    :NX$=CHR$(186)+"NX:"
    :BA$=CHR$(95)
210 AD=80:SC=32768:Z$=CHR$(0)
    :LF=10:QT=34:POKE 59468,14
220 SS=80:POKE SC+1024,42
    :IF PEEK(SC)=42 THEN SS=40
230 INPUT"FROM ASCII FILE";FF$
240 INPUT"FROM DRIVE";FD$
250 INPUT"TO DRIVE";TD$
260 INPUT"MAX WORDPRO LINES PER
    OUTPUT FILE";MX$
270 MX=INT(VAL(MX$))
    :IF MX<5 OR MX>330 GOTO 260
280 CLOSE 15:OPEN 15,8,15
    :PRINT#15,"I"FD$:GOSUB 430
    :PRINT#15,"I"TD$:GOSUB 430
290 CLOSE 1:OPEN 1,8,3,FD$+"
    :"+FF$+",S,R":GOSUB 430
300 N$=FF$:IF RIGHT$(N$,
    1)="*"THEN N$=LEFT$(N$,
    LEN(N$)-1)
310 N$=LEFT$(N$,10):F=1:GOSUB 450
320 CLOSE 2:OPEN 2,8,4,TD$+"
    :"+TF$+",P,W":GOSUB 430:LN=0
330 PRINT#2,CHR$(AD)CHR$(AD);
    :I$=CM$+TF$:GOSUB 390
340 GET I$:SYS 896:SW=I$="Q"OR ST
    :IF SW THEN CLOSE 2:CLOSE 1
    :CLOSE 15:END
350 L=PEEK(139):I$=LEFT$(A$,L)
360 IA=ASC(I$+Z$):IF LF=IA OR IA
    =QT THEN I$=MID$(I$,2)
    :GOTO 360
370 GOSUB 390:IF LN<MX-1 GOTO 340
380 F=F+1:GOSUB 450:I$=NX$+TF$
    :GOSUB 390:GOTO 320
390 PRINT CHR$(147)I$BA$
    :L=LEN(I$)
400 CS%=L/SS+1:FC=CS%*SS-L-1
    :LN=LN+CS%
410 FOR I=0 TO L:PRINT#2,
    CHR$(PEEK(SC+I));:NEXT I
    :GOSUB 430
420 IF FC THEN PRINT#2,LEFT$(BL$,
    FC);
430 INPUT#15,E,E$:IF E THEN PRIN
    T E;E$:CLOSE 15:END
440 RETURN
450 TF$="W."+RIGHT$("00"+MID$(ST
    R$(F),2),2)+N$:RETURN
460 REM----------
```


COMMODORE 64™

INSULATING?

Energy prices are going to continue to rise. That's why many people are taking a close look at their homes to see where they can save on heating bills. The problem is where to begin. Do you add more insulation? Where? Buy a new furnace? Or maybe add some thermal shutters to the windows?

THE FACT IS THAT MOST PEOPLE DON'T KNOW WHERE THEY ARE LOSING THE MOST HEAT.

St. Croix Valley Electronics has developed the Energy Analysis Program to assist you. You simply answer questions about your existing insulation, house dimensions, type of furnace, and type of house construction, and the program computes the rest. You'll find out how much heat you're losing and where, and also the amount of money saved by making a given improvement. You also find out how long it will take for the improvement to pay for itself.

Specify tape or diskette when ordering.
PRICE: $26.64 Cash, $29.00 VISA, MASTERCARD

St. Croix Valley Electronics
P.O. Box 75113
St. Paul, MN 55175

Wisconsin residents please add 5% state sales tax.
Commodore 64 is a Trademark of — Commodore Electronics Ltd.

Delayed Shell Sort

by Joe Rotello

Nothing makes believers out of computer agnostics quite like seeing how fast the computer can sort data. Fortunately (?), there are probably as many sorts as there are needs for them, with names like Bubble, Shell, Quick and Insertion.

One of the most popular sorts for micros has been the Shell Sort, named, quite aptly, for its inventor, Donald Shell. It has been a simple but reliable workhorse for many years but, like all horses, improvements can be and are made. The Shell-Metzner sort is one such improvement and another improvement is what we want to discuss here.

A note of caution: We do not intend to dissect or teach the innards of the sort routine presented here any more than we have to. Our main purpose is to present an improved sort routine that can be put to actual use right from the book, so to speak.

The improvement results in a program called a Delayed Shell Sort (DSS for short). It is actually an improved Shell-Metzner routine that we will not only use to sort an alpha string (name, address, etc.) but will assign a useful task as well, aside from sorting just a string. We will "tack on" a numeric pointer that will, in this case, represent a relative file record number that is associated with its respective alpha string. We now have a useful sort routine, improved substantially performance-wise and at the same time, given a "real life situation" use and function.

The main improvement we have made in the sort routine is to delay, as much as possible, any exchange of elements for as long as possible. In BASIC, it is the constant exchange of data that seems to be the mortal enemy of most sorts. So it should follow that if extra sort work is kept to a minimum, a sort routine should end up that much more efficient which, in BASIC, translates to that much extra speed.

In comparison to the "standard" Shell or Shell-Metzner BASIC sort, the DSS sort routine is about 25% faster, in terms of overall execution speed. In fact, if you remove the BASIC code reference to the relative record pointer, the DSS sort can be as much as 30% faster than its famous parents.

Chart 1 describes the results of our test runs using the DSS with the PETSpeed BASIC compiler. You can see that the sort can be speeded up even more by compiling under PETSpeed. In fact, when you note the excellent compiler sort times in Chart 1 you will begin to understand why PETSpeed is really catching on in the PET/CBM/64 programming world!

Since we have to start somewhere, we built a short program to automatically create random strings composed of user-determined byte lengths. Hence, you can test the sort over a wide area of string quantity and string length with the program. As promised, each string is given a integer pointer that will mimic a relative file pointer (or any other number, really) in the sequence that the string is produced. That is, string number one output from the create routine will be assigned "pointer" one and string number two "pointer" two and so on.

The program is set to allow up to 500 entries of up to 25 bytes in length each, but with 32K (or more) memory, you could sort many more items if you had a need to. 500 just seemed to be a logical top end for our discussion, as did 25 bytes as a logical maximum length.

The Once Over

The program works as follows.

Lines 60-90 Dimension our variables and input the quantity of strings to sort (TW).

Line 100 "Zeros" the system clock and sets variable counter N.

Lines 130-240 The heart of the DSS sort routine. Note that the "pointer" (pseudo relative file record number) is sorted along with its associated string as P%(I). Although the DSS routine has three distinct IF...THEN logic calls, the overall performance of the routine exceeds that of the "standard" Shell or Shell-Metzner routines.

Lines 260-280 Display the actual sort time (in seconds) and print, in sorted order, the index number and the sorted string with its relative record number.

Lines 320-380 Comprise our string-create routine. The user enters the requested string byte length in the range of 1-25. Lines 340-360 build the random byte string from the 26 allowable letters of the alphabet, each being length Q. Line 360 assigns each created string its own place in the string array A$(I). Line 370 assigns each string its own "relative record number" as P%(I), increments I for the next loop and exits the loop if the requested TW number of strings has been formed.

Customizing

The DSS sort routine can be customized for disk work by eliminating the code in lines 60-100 and 310-380. Replace those lines with your own disk input routines making sure that A$(I) holds the string to be sorted and P%(I) holds the relative record number (or other numeric value) associated with that string.

The sort works on strings of varying lengths. Our little "automatic string" generator creates all strings of the same length, but that was just for purposes of generating a bunch of example strings quickly. Any string length between 1 and 255 can be sorted and the data to be sorted can be composed of any mixture of string lengths.

The sort as presented should work fine on any Commodore product, just be sure that you allow for sufficient memory space to hold both the sort and whatever program you will be using with the sort.

Last but not least, take advantage of the speed power of the PETSpeed compiler. If you can use PETSpeed on this sort routine, by all means do so. PETSpeed can and will enhance your programming efforts. C

Chart 1.
Averaged Sort Times for DSS BASIC and DSS PETSpeed

(Example string length = 10 characters)

No. of Items	DSS BASIC	DSS PETSpeed
25	2.8 sec	.30 sec
50	7.2 sec	.75 sec
100	19.1 sec	1.9 sec
250	61.0 sec	6.4 sec
500	149 sec	14.0 sec

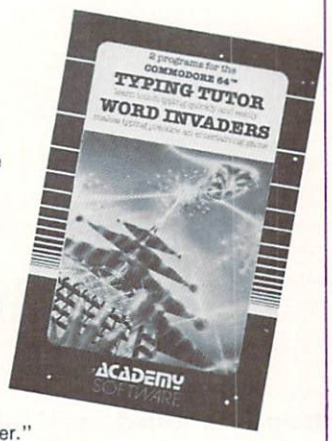
NOTE: Sort times include BOTH string AND relative record pointer being sorted in one operation.

Shell Sort

```
10 REM     DPOINTSORT                    50 :
20 REM     DELAYED SHELL SORT            60 DIM A$(500),P%(500)
30 REM     WITH REL RECORD POINTER          :PRINT"[CLEAR,UP]";
40 REM                                   70 INPUT"# OF STRINGS (500 MAX)
```

VIC 20™ COMMODORE 64™

Still the Best!



Rated **THE BEST** educational program for the VIC 20™ by ***Creative Computing*** magazine.

Commodore 64 version: "This is the best typing tutor we have seen yet; it can get your children touch typing in short order and bring an old hand up to speed. Includes excellent training modules and an arcade type mode to liven things up and put some pressure on; ★★★★+" **INFO-64**

Our customers continue to tell us of their success. . . .

". . delighted with my son's progress . . . he is the only one in his second grade class who touch types at the computer."

(58 year old man writes) . . . "great, excellent. To me a source of great learning . . . I just can't express how much I have enjoyed it!"

In daily use by schools across the USA.

"Computer aided instruction at its best" ***Commander*** magazine

TYPING TUTOR + WORD INVADERS

The proven way to learn touch typing.

COMMODORE 64 Tape $21.95
COMMODORE 64 Disk $24.95
VIC 20 (unexpanded) Tape $21.95

NEW!

IFR
(FLIGHT SIMULATOR)

DISK OR TAPE FOR THE COMMODORE 64 $29.95

CARTRIDGE FOR THE VIC 20 $39.95
JOYSTICK REQUIRED

Put yourself in the pilot's seat! A very challenging realistic simulation of instrument flying in a light plane. Take off, navigate over difficult terrain, and land at one of the 4 airports. Artificial horizon, ILS, and other working instruments on screen. Full aircraft features. Realistic aircraft performance—stalls/spins, etc. Transport yourself to a real-time adventure in the sky. Flight tested by professional pilots and judged "terrific"! Rated "Excellent" by Midnite Software Gazette.

VISA Shipping and handling $1.00 per order. CA residents add 6% tax. MasterCard

ACADEMY SOFTWARE

P.O. Box 6277, San Rafael, CA 94903 (415) 499-0850

Programmers: Write to our New Program Manager concerning any exceptional VIC 20™ or Commodore 64™ game or other program you have developed.

user departments:

PET/CBM

```
   TO SORT ";TW
80 IF TW>500 THEN PRINT"[UP]";
   :GOTO 70
90 GOTO 320
100 PRINT:PRINT CHR$(7);
    "SORTING "I-1" ITEMS":N=I-1
    :TI$="000000"
110 :
120 REM   DELAYED SHELL/METZNER
130 D=2^INT(LOG(N)/LOG(2))-1
140 FOR I=1 TO N-D
150 IF A$(P%(I))<=A$(P%(I+D))THE
    N 230
160 T%=P%(I+D):P%(I+D)=P%(I)
170 IF I<=D THEN P%(I)=T%
    :GOTO 230
180 FOR J=I-D TO 1 STEP-D
190 IF A$(T%)>=A$(P%(J))THEN 220
200 P%(J+D)=P%(J)
210 NEXT J
220 P%(J+D)=T%
230 NEXT I
240 D=INT(D/2):IF D>0 THEN 140
250 :
260 V=TI/60:FOR N=1 TO I
270 PRINT N,A$(P%(N)),P%(N):NEXT
280 PRINT V" SECS SORT TIME FOR
    "I;" ITEMS"
290 END
300 :
310 REM    STRING CREATE ROUTINE
320 I=1
330 INPUT"BYTE LENGTH (1-25) ";Q
335 IF Q>25 THEN PRINT"[UP]";
    :GOTO 330
340 FOR K=1 TO Q
350 C$=CHR$(INT(RND(1)*26+65))
360 B$=B$+C$:NEXT K:A$(I)=B$
    :B$="":PRINT I,A$(I)
370 P%(I)=I:I=I+1:IF I<=TW THEN
    340
380 N=I:GOTO 100
```

SuperPET Potpourri

by Dick Barnes,
Editor, SuperPET Gazette

This issue we cover SuperPET's control codes and character set and how to issue DOS commands wherever the microEDITOR is used (all languages but APL). You may search the manuals for details on both—in vain. So far as I know, the summaries below have never before appeared in print. In the last part of this column, I present some very useful but more advanced goodies.

One important note on all SuperPET material: all languages are case-sensitive. When I use lower case, do not substitute CAPITALS and vice-versa. All programs are normally written in lower case and should be so copied.

Character Set and Screen Control

The Roman character set in SuperPET is pure ASCII (not PET ASCII). Any ASCII table defines the printable character set and the CONTROL functions of the non-printable codes. If you add 128 to its ASCII code, you can print any character to screen in reverse field. POKE, PEEK and print codes, wherever used, are identical for a given character or control function.

The fourteen CONTROL codes summarized below apply to SuperPET's screen, screen printing, disk files and general operations, in all languages including APL (in APL, quad IO, or index origin, must be zero for the codes to work. APL boots with IO set to one. Allow for that.)

ASCII CODE:	KEY TO USE/COMMENTS:	SCREEN EFFECT:
1	HOME. HOME key or in program.	HOMEs the Cursor
2	RUN. SHIFT/RUN in microEDITOR starts any program.	None
3	STOP. STOP key *pauses* program; in program, ends execution. ASCII 3 in disk file stops a read in V1.0 software only.	None
4	DELETE key deletes character the cursor is on, pulls characters from the right.	DELETE right.
5	Usual Commodore SHIFT/INSERT.	INSERT space
6	Use ESCAPE key or in program.	ERASE from cursor to end line
7	CURSOR key and in program.	CURSOR right
8	SHIFT/CURSOR key and in program.	CURSOR left
9	TAB key tabs to default or user-set tabs. Useful in program.	TAB to next preset stop.
10	CURSOR key and in program.	CURSOR down
11	SHIFT/CURSOR key and in program.	CURSOR up
12	SHIFT/CLR key and in program.	CLEAR screen, home cursor
13	RETURN key and in program.	CARRIAGE return
127	REPEAT key (old Commodore DELETE).	DELETES character to its left

(All SuperPET keys repeat, so you don't need the old REPEAT key function.)

You'll find useful an old Commodore trick with command codes when writing a program. Any such code, like a carriage return or tab, is best written as a string constant, since CR$ runs a lot faster than chr$(13), is shorter to write and tells you when you see it what it does. Define such control strings at the start of a program, as I've done below. Put command strings in upper case, so you can spot 'em fast; leave the rest of the code in lower case (why shift?).

If you've previously controlled the cursor with quotation marks and reverse-field symbols, you'll find SuperPET code much simpler to read. Here is an example: we clear screen, print a line, double-space to the next line and then replace what we printed with another line. See how the command strings, in caps, stand out.

```
90 ! Home Cursor    Cursor Down     Erase Line      Clear Screen
100  H$=chr$(1)   : D$=chr$(10)  : El$=chr$(6)   : CS$=chr$(12)
```

```
190 print CS$; "Double space
      after this line."; D$
200 print H$; EL$; "I like this
      line better."
```

Don't jam your code together without spaces for speed. The folks at Waterloo made it unnecessary. If you remove *all* spaces from an mBASIC program, you decrease run time only 2.6%—hardly worth the wear and tear on thine eyeballs.

DOS Commands

While you may send immediate-mode DOS commands from any language, it's simplest to use the version of the microEDITOR (mED) which runs in that language (or in the mED loaded alone). Load any language but APL or the mED alone with a simple: e RETURN and try some of the commands that follow with scratch disks, from command mode. You get that mode in the mED by touching PF 5 (the phrase means SHIFT KEY-PAD 5, or "Programmed Function" key 5). You'll find two cursors on the screen in command mode. The lower, at line 24, is the command cursor, and that's where we enter all DOS commands from the keyboard. (Return to screen mode with PF 8 when you're through.)

SuperPET uses two sets of DOS commands; those from BASIC 3.0, and its own intrinsic commands. Though the 3.0 DOS commands are defined in CBM manuals, some don't apply and the format is changed. Table I shows all the 3.0 commands you need to know, shown for drive one as destination. Invert examples for drive zero.

Now that we have the commands, how do you use them? What follows would be a lot harder if a Canadian named Jim Swift had not discovered (delightfully) that you can execute any 3.0 DOS command if you preface it with: g ieee8-15. This "get" opens the command channel (15) to our drives (device eight). You may find it strange to use a "get" to "put" a DOS command (and Waterloo did *not* design the system that way)—but it works! Put a scratch disk in drive zero and I'll prove it. Enter the command below, exactly as given. Then hit RETURN.

```
g "ieee8-15.NO:test file,te"
(you can substitute ' for ")
```

Note the quotes, used only when there's a space in the filename. When the disk is formatted, try a *fast* way to scratch any disk, no

TABLE I. 3.0 DOS Commands for SuperPET. Destination drive is left of equal sign.

The DOS command (such as D, below) must be in CAPITAL letters.

D1=0	DUPLICATE. Creates backup on disk 1 of disk 0.
C1=0	COPY. Copies to disk 1 all of disk 0.
N1:diskname,id	NEW. Formats disk 1 with title "diskname", and id of "id"
C1:big=0:tiny,1:teeny,0:weeny	CONCATENATE. Creates file "big" on drive 1, composed of files "tiny, teeny, and weeny"
V1	VALIDATE disk 1. (Same as 4.0 COLLECT)
C1:newfile=0:oldfile	COPY to disk 1 file "oldfile" from drive 0 and renames it to "newfile".

matter how full. Leave the disk you just formatted in drive zero and enter (without quotes):

g ieee8-15.NO:trials
(Don't enter an ID number or a comma!)

When you hit RETURN, you'll have a scratched but formatted disk with a new title in three seconds flat—or less.

Use any 3.0 DOS command listed above with the preface of: g ieee8-15. Two minor problems may arise. First, the DOS commands are limited to 40 characters, and may fail if you try to manipulate a "verylongfilename" The solution is simple. Copy or concat to a new, short filename, as shown below:

C1:x=0:verylongfilename.
(Rename the new file "x" later.)

The second problem is a rare, occasional bug in: g ieee8-15. For reasons no one has discovered, it may not close channel 15 after it has executed the DOS command. You wait, but the red LED on the drive doesn't go off. Solution: hit STOP. A screenful of '00 , OK,00,00' scrolls into view; the disk file closes. Delete the OK's, and no harm is done to your disk or screen file. So long as you know it can happen, and how to cure it, it is no problem.

We've covered all the 3.0 DOS commands you need and pass to the built-in SuperPET DOS commands. None need the g ieee8-15. prefix. Give the commands as written below. Remember that drive zero, unit eight, is the default drive.

Be careful. If you copy files, I recommend you use the 3.0 DOS command, for it will *always* copy a file in its original DOS format, without error. Yet the SuperPET COPY command will let you copy files to and from any disk devices, whatever their device numbers. The g ieee8-15. method will not (onu9, however given, for example, fails to work). So use the COPY

INTRINSIC COMMAND:	SHORT FORM, AS GIVEN:	WHAT IT DOES:
MOUNT	mou disk	Initializes drive 0.
COPY *(and rename)*	cop disk/1.title to title	Copies file "title" from disk 1 to disk 0.
	cop oldname to disk9/1.newname	Copies file "oldname" on disk 0 to disk 1 on unit 9.
	cop faster to faster.bak	Creates a backup of file on disk 0.
SCRATCH	scr disk/1.badfile	Scratches "badfile" from disk 1.
	scr *	Scratches all of disk 0, the default drive.
RENAME	ren test to tested	Renames file "test" to "tested" on disk 0.
	ren disk/1.test to tested	The same on disk 1.

COPY requires the DOS format of the file but defaults to SEQ if it is not stated. All other DOS formats (USR, REL, PRG) must appear in the COPY command, as shown below.

SOME COPY COMMANDS:	COMMENTS:
cop disk/1.example,prg to example,prg	PRG must appear on both the source and destination filenames.
cop disk9/1.test,rel to disk/1.test	ERROR. The DOS format REL is not stated in the destination. The file will be copied as a SEQ file!

method best suited for the job.

One last warning. Improperly closed files show on a directory with an asterisk preceding the DOS format (as in *SEQ). *Never* scratch such a file. Instead, get rid of it (assuming it's on drive 1) with a VALIDATE: g ieee8-15.V1 RETURN.

A Few Goodies

There's an occasional bug in 8050 drives, which won't let you read or write to disk at times. You try to get a file or a directory and get a DRIVE NOT READY message (and if you try again, an IO TIME OUT), even though the drive is on and disks are in both slots. It seems to happen after you've accessed a track larger than 55 and have then shut down. On the next power-up, you get the bug.

Jim and Ellen Strasma, in the *CBM Professional Computer Guide,* recommend that you quickly open and close the drive doors while the drive attempts to read a directory; after lots of error lights and some awful noises, according to John Frost of Seattle, you get an error message. Enter another directory request and the bug is cleared—sometimes. It worked for me once and then failed the second time. By trial and error I found out that the MOUNT command, given from the mED after I loaded it with my 4040, *always* kicked the 8050 drives back into operation. But how do you load the mED (or anything else) if you have only one drive and it won't work? It took me two weeks to figure it out: you use the monitor, which you can enter from menu with: m RETURN. So, here's the monitor code to generate MOUNT for both drives zero and one (the beginning "m" means "modify"):

```
>m 1000 cc 10 07 bd b0 e7 3f
64 69 73 6b 2f 30 00 RETURN
>g 1000 RETURN
```

The code above is for drive zero. Code for drive one is identical except for the last two bytes. Change them to: 31 00 for drive one and do the "go 1000" again. Both drives are initialized—and start working. We've used the method twice, and it has worked both times.

Our second goodie is a RESET, which takes you from main menu *back* to whatever language or facility you had previously loaded in the upper 64K of bank-switched memory. Suppose you have left 6809 for 6502 mode and then come back to 6809. Do you want to twiddle your thumbs while you reload APL? No? Well, use RESET. In about three seconds, you have APL again (or whatever else you previously had in the upper 64). The same RESET works if you accidentally leave a language and want it back or when you purposely leave a language to load a machine-language program from main menu. The code is generated by the microBASIC program shown in listing one, which puts to disk a machine-language module named "reset". Once you have the module, throw the mBASIC program away. Copy "reset" to your language disk after you've tested it and load it from disk one any time you want to RESET with a simple: reset RETURN (the menu loader will look for it on drive one).

Terry Peterson of El Cerrito, California, wrote the code, which most cleverly loads *absolutely nothing* but tricks SuperPET into thinking something *was* loaded, when, in fact, all that gets turned on is the interpreter or facility that already resides in the upper 64. You don't even need to put the READ/WRITE switch in READ. Why bother? Nothing is loaded.

Of course, if you've used the upper 64K whilst in 6502 mode, the program will not work. Normally, you don't use it, so RESET is very valuable. (For more information on SuperPET, write: Editor, SuperPET Gazette, PO Box 411, Hatteras, N.C. 27943.) **C**

Listing 1: RESET:BU

A microBASIC utility program which creates a machine-language module to reset SuperPET to any language or facility resident in the 64K of bank-switched memory.

```
100 ! reset:bu : An mBASIC program to create a load module which
110 ! will restart any of the WCS interpreters from main menu.
120
130 open #9, 'reset,prg', output
140 print #9, chr$(hex('90')); rpt$(chr$(0),3); chr$(1); rpt$(chr$(0),5);
150 print #9, chr$(2);chr$(0);
160 close #9
170 ! Written by Terry Peterson.
```

A Closer Look at the B Series:
The Machine Language Monitor

by Howard Rotenberg
Commodore Canada

Howard continues his introduction to Commodore's newest "business" system.

At this point it is time to look at the B Series computer in a little more depth. A good place to start is with the machine language monitor. The new monitor has all the old commands plus six new ones, although the syntax of some of the old commands has changed to fit the new computer. We will look at the new commands at this point.

TIM is the terminal interface monitor program for MOS Technology's 65XX microprocessors. It has been expanded and adapted to function on the B Series computers. Execution is transferred from the CBM BASIC interpreter to TIM by the SYS command. The monitor is incorporated as part of the Kernal.

Commands typed on the CBM keyboard can direct the TIM to start executing a program, display or modify registers and memory locations, load or save binary data, view other segments, send disk commands or read status, set default disk unit and load and execute programs by entering the program name (segment 15 only). On modifying memory, TIM *no longer* performs automatic read after write verification to insure that the addressed memory exists, and is R/W type.

TIM Commands

M Display memory
: Alter memory
R Display registers
; Alter registers
G Begin execution
L Load
S Save
V View segment
U Set default disk unit
@ Send disk command or get disk status
X Exit to basic
Z Transfer to second microprocessor
(file name) load and execute

Examples

M Display Memory

```
M 0000 0010
: 0000 0f 0f 4c d9 9a 00 00 00 00 00 00 00 22 22 9e 00
: 0010 00 00 00 00 00 00 00 d4 fb 04 00 04
  00 00 c4 fb
```

In a display memory command, the start and ending addresses must be completely specified as four-digit hex numbers. To alter a memory location, move the cursor up in the display, type the correction and press RETURN to enter the change. When you move the cursor to a line and press RETURN, the colon tells the monitor that you are re-entering data.

R Display Registers

```
R
   PC IRQ SR AC XR YR SP
; 0007 FBF8 B0 DD 71 04 71
```

The registers are saved and restored upon each entry or exit from the TIM. They may be modified or preloaded as in the display memory example above. The semicolon tells the monitor you are modifying the registers.

G Begin Execution

```
G 0200
```

The GO command may have an optional address for the target. If none is specified, the PC from the R command is taken as the target.

L Load

```
L "filename", 08
```

No defaults are allowed on a load command. The device number and the file name must be completely specified. Operating system prompts for operator intervention are the same as for BASIC. Memory addresses are loaded as specified in the file header, which is set up by the SAVE command. Machine language subroutines may be loaded from

BASIC but care must be taken not to use BASIC variables as the variable pointer is set to the last byte loaded plus one. The machine language subroutine will be loaded into the segment that you are currently in as determined by the V command. After the load, the system will be initialized back to segment 15.

S Save

S "filename", 08,010200,010300

As in the load command, no defaults are allowed in the SAVE command. The device number, file name and a six-byte start and end address must be given. The above example will save a program to device eight from segment number one starting at 0200 hex and ending at 0300 hex. The first two bytes are the segment number followed by the address. Valid segment bytes may be 00 to 0F depending on your memory. After a save, the system will be initialized back to segment 15.

V View

V 01

This will change the segment to the one that you wish to view, save, load or change memory from. The valid segments are 00 to 0F.

U Unit Address

U 09

This command will allow you to set the disk unit default address while you are in the monitor. When you leave, the original address is reset. Valid unit addresses are 8 to 1F. These must be entered in HEX.

@ Read Error Channel and Process Disk Commands

@	:Display error message and clear channel
@ S1:filename	:Scratch specific file from drive 1
@ I0	:Initialize disk in drive 0
@ R0:newname = oldname	:Rename file on drive 0
@ C1:filename = oldname	:Copy file from drive 0 to drive 1
@ V0	:Validate or collect disk in drive 0
@ N1:filename,id	:New or header disk in drive 1

The above examples use the same syntax as the wedge program supplied with the disk drives.

(File name) Load and Execute File in Segment 15

This will load a machine language program from the disk and execute it. Its use is restricted to segment 15.

Z Transfer to Second Microprocessor

Z

This command will allow you to utilize the auxiliary processor when applicable.

X Exit to BASIC

X

This will cause a warm start to BASIC. In a warm start, memory is not altered in any way and BASIC resumes operation the way it was before the call to the monitor was made.

In my next article I will discuss parts of the memory management and more information about the computer in general. Until then, B seeing you. **C**

Programming B128 Function Keys

by Al Fragola

There are 20 available function keys on the new Commodore Model B128-80 computer. The keytops are labeled F1 through F10. Keys F11 through F20 are obtained by holding down the SHIFT key while pressing the appropriate function key, so that shifted F1 yields F11. Upon power-up, the first ten functions are pre-programmed. Keys 11 through 20 are not programmed.

The result of pressing a function key is that a given string of characters is printed to the screen, with the cursor residing immediately to the right of the printed material. As pre-programmed, no carriage return is printed, allowing you to add whatever text or command is desirable. Thus, if you press F2 and RETURN, you will get a complete program listing to the screen.

Obviously, there are ten unused keys for your convenience, as well as the ability to re-program F1 through F10 to suit your own personal tastes. This is a relatively simple process, and it is then possible to save several sets of function key images on disk for use in various tasks or in conjunction with different software packages as pre-programmed responses. Thus you could have a programming assistance package or a graphics package, as you see fit.

Before we look at how these keys are re-programmed, let's first look at where and how the B128 stores the function key information. The key characters are stored one following the other in the upper memory of bank two, starting at decimal 64512. No delimiters are placed between each string. Instead, the length of each function key string is stored in bank 15, at decimal 899 through 918, using one byte for each function key. This causes a slightly awkward situation, since to change one pre-programmed key, the new string must occupy exactly the same number of bytes. If the new string is shorter than the old one, you can pad with spaces. If it is longer, however, the remaining strings must be shifted to make room. For the sake of simplicity, we will only look at replacing the entire set of function keys.

The Program

The simplest approach is to set up the various strings as variables and POKE their ASCII values into the appropriate memory locations. Once this is done, their lengths must be POKED into bank 15. Later, we will look at a program for retrieving the new function key strings. Refer to Program 1.

Line 100: Sets up arrays FU$ and FL to capture the function key strings and their lengths.

Line 110: Sets NF equal to the number of keys you wish to program.

Line 120: Determine here whether you wish to start at F1 or F11. For the sake of simplicity, no attempt has been made in this program to start within the existing pre-programmed keys.

Lines 140-170: Inputs the new strings into the array FU$(I) and their lengths into FL(I)

Line 180: Selects bank 2 for PEEK/POKE use

Line 190: Sets us up for F1-F10 or F11-F20. Note that ELSE is now available in BASIC.

Lines 200-250: POKEs the ASCII values of the strings into the appropriate locations.

Line 260: Same as 190, but for lengths.

Line 270: Selects bank 15 to POKE in lengths.

Lines 280-300: POKEs lengths into appropriate addresses in bank 15.

Line 310: POKEs zeros into the length addresses of unused keys

Now we need a method to store the new function key strings on disk. Fortunately, the B128 has handy commands to load (BLOAD) and save (BSAVE) binary information to and from memory. It will take two operations to save the information placed into Banks 1 and 15. LINES 320 through 370 perform this task for you.

Lines 320-340: Asks if you wish to save your formats. If not, the program ends.

Line 350: Sets fi$ equal to the name of the new key formats.

Lines 360-370: BSAVES the two memory blocks involved, and coordinates file retrieval by suffixing one file with ".1" and the other with ".2". Note that the use of variables in BSAVE requires that they be in parentheses.

To load your new set of function key strings, simply load and run Program 2. It asks for the name you have given the format set and then loads the two parts into the appropriate Bank. Program 2 can be incorporated into other programs as a subroutine to call in new sets of function keys as you need them. It does not disturb the BASIC program that is running.

Using this little package, you can set up several varieties of function key arrangements that suit your specific needs at varying times. There is one limitation that I have found. The total number of characters in the function key group *cannot* exceed 1024. That represents ALL the memory above 64512. Also, this is based on my experience with the B128. So far, I have found no other uses of this block of memory, but I may be wrong. **C**

Al Fragola teaches business management at Tarrant County Junior College in Hurst, Texas.

Program 1.

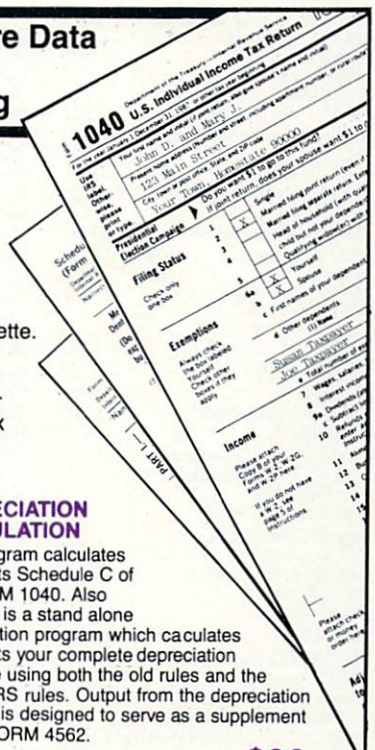
```
100 DIMFU$(20),FL(20)
110 INPUT"HOW MANY FUNCTIONS";NF
120 INPUT"START AT (A) F1  OR  (B) F11";SA$
125 IFSA$="B"THENX=10
130 FORI=1TONF
140 PRINT"STRING FOR KEY F"I+X;
150 INPUT FU$(I)
155 INPUT"END WITH CARRIAGE RETURN  Y/N";Z$:IFZ$="N"THEN160
156 IFZ$="Y"THENFU$(I)=FU$(I)+CHR$(13):ELSE155
160 FL(I) = LEN(FU$(I))
170 NEXTI
180 BANK 2
190 IF SA$="A" THEN SA = 64511:ELSE SA=64568
200 FOR I = 1 TO NF
210 FOR J = 1 TO FL(I)
220 POKE SA+J,ASC(MID$(FU$(I),J,1))
230 NEXTJ
240 SA = SA + J -1
250 NEXT I
260 IF SA$ ="A" THEN SB=898:ELSE SB=908
270 BANK 15
280 FOR I = 1 TO NF
290 POKE SB+I,FL(I)
300 NEXT I
310 IFSA$="A"ANDNF<20THENFORI=899+NF TO 919:POKE I,0:NEXTI
320 PRINT"DO YOU WANT TO SAVE THIS ON DISK?  Y/N"
```

```
330 GETZ$:IF Z$<>"Y" AND Z$<>"N" THEN 330
340 IF Z$="N" THEN END
350 INPUT"NAME THIS FILE (10 CHAR MAX)";FI$
360 BSAVE(FI$+".1"),B2,P64512 TOP(SA)
370 BSAVE(FI$+".2"),B15,P899 TOP 918
380 END
```

Program 2.

```
10 input"which function key file";fi$
20 bload(fi$+".1"),b2,p64512
30 bload(fi$+".2"),b15,p899
```


SUPERTAX™ Get Supertax by Rockware Data and get the jump on your 1984 Income Tax Planning



THIRD SUCCESSFUL YEAR! • THOUSANDS ALREADY IN USE!

Use SUPERTAX personal income tax programs to calculate your tax liability now and have plenty of time to make year-end investment decisions to improve your position. SUPERTAX was specifically created for Commodore 64 users by a practicing CPA with a Master's degree in tax accounting. Highly acclaimed by tax pros, CPA's and tax preparers, SUPERTAX is easy to understand and a pleasure to work with.

- SUPERTAX is fully screen-prompted and includes a manual loaded with valuable tax information, instruction and guidance.
- SUPERTAX instantly recalculates your entire return when you change any item.
- SUPERTAX prints directly on IRS forms.
- SUPERTAX DATA can be stored on a diskette.
- SUPERTAX updates are available at 50% discount to registered SUPERTAX owners.
- SUPERTAX is an essential addition to your personal software library—best of all it's tax deductible.

FOR TAX PLANNING

Using either screen or printer output, SUPERTAX generates clear and concise summaries of Page 1 and 2 and Schedule A of FORM 1040 allowing you to see at a glance and to quickly comprehend your tax situation. This program also prints an OVERALL SUMMARY of the return showing Adjusted Gross Income, Itemized Deductions, Taxable Income, Regular Tax and Payment Due or Refund—all of which are calculated by the program. SUPERTAX also calculates the moving expense deduction, investment credit, taxable capital gains, political and child care credits, medical limitations, and much more. Input is fast and easy and changes can be made in seconds. This program actually makes tax planning a breeze.

FOR RETURN PREPARATION

SUPERTAX PRINTS THE INCOME TAX RETURN: This program prints page 1, page 2 of the FORM 1040, Schedules A, B, W and G (income averaging) of the FORM 1040 as well as FORM 3468 (investment tax credit) on standard IRS government forms or on blank computer paper for use with transparencies. Any item of input can be changed in seconds and the entire return is automatically recalculated to instantly reflect the change.

NOTE: Printing on government forms requires friction feed printer.

FOR DEPRECIATION CALCULATION

This program calculates and prints Schedule C of the FORM 1040. Also included is a stand alone depreciation program which caculates and prints your complete depreciation schedule using both the old rules and the new ACRS rules. Output from the depreciation program is designed to serve as a supplement to IRS FORM 4562.

Products shipped FOB Dallas, TX
Commodore 64 is a trademark of Commodore Business Machines

$99

TO ORDER Call Toll Free 1-800-527-4171 In Texas Call 214-739-1100
MasterCard, VISA, Money Orders, Bank Checks and COD Orders Accepted
(add 3% surcharge for credit card processing) (add $5.00 for COD) ($3.00 Shipping)

For Brochure WRITE
Financial Services Marketing Corp.
10525 Barrywood Drive
Dallas, Texas 75230

user groups

User Group Listing

ALABAMA

Huntsville PET Users Club
9002 Berclair Road
Huntsville, AL 35802
Contact: Hal Carey
Meetings: every 2nd Thursday

Riverchase Commodore Users Group
617 Grove St.
Birmingham, AL 35209
(205) 988-1078
Ken Browning

Wiregrass Micro-Computer Society
Commodore SIG
109 Key Bend Rd.
Enterprise, AL 36330
(205) 347-7564
Bill Brown

Tiger Byte: E. Alabama CBM 64 Users Group
c/o The Computer Store, Inc.
Midway Plaza
Opelika, AL 36801
Jack Parsons
1st & 3rd Wed. of Month

The Birmingham Commodore Computer Club
Birmingham, AL
(205) 923-9260
Harry Jones

Commodore Club of Mobile
3868-H Rue Maison
Mobile, AL 36608
(205) 343-1178
Tom Wyatt
3rd Thurs. of month

Shoals Commodore Users Group (SCUG)
209 Lakeshore Dr.
Muscle Shoals, AL 35661
Geo. Taylor
2nd & 4th Tues. of month

1920-A Avenue C
Brookly
Mobile, AL 36615
(205) 661-1973
Howard Crider

ALASKA

COMPOOH-T
c/o Box 118
Old Harbor, AK 99643
(907) 286-2213

Alaska 84 Computer Club
c/o Line 49 Management
P.O. Box 6043
Anchorage, AK 99502

First City Users Group
P.O. Box 6692
Ketchikan, AK 99901
(907) 225-5695
James Llanos

ARIZONA

VIC Users Group
2612 E. Covina
Mesa, AZ 85203
Contact: Paul Muffuletto

Catalina Commodore Computer Club
2012 Avenida Guillermo
Tucson, AZ 85710
(602) 296-6766
George Pope
1st Tues. 7:30 p.m.

Central Arizona PET People
842 W. Calle del Norte
Chandler, AZ 85224
(602) 899-3622
Roy Schahrer

ACUG
c/o Home Computer Service
2028 W. Camelback Rd.
Phoenix, AZ 85015
(602) 249-1186
Dan Deacon
First Wed. of month

West Mesa VIC
2351 S. Standage
Mesa, AZ 85202
Kenneth S. Epstein

Arizona VIC 20-64 Users Club
232 W. 9th Place North
Mesa, AZ 85201
Donald Kipp

Arizona VIC & 64 Users
904 W. Marlboro Circle
Chandler, AZ 85224
(602) 963-6149
Tom Monson

ARKANSAS

Commodore/PET Users Club
Conway Middle School
Davis Street
Conway, AR 72032
Contact: Geneva Bowlin

Booneville 64 Club
c/o A. R. Hederich Elementary School
401 W. 5th St.
Booneville, AR 72927
Mary Taff

The Siloam Commodore Computer Club
P.O. Box 88
Siloam Springs, AR 72761
(501) 524-5624
Ken Emanualson

Russellville Commodore User Group
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

Arkansas River Valley Commodore Users
401 S. Arlington Dr.
Russellville, AR 72801
(501) 967-1868
Bob Brazeal

CALIFORNIA

SCPUG Southern California PET Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: First Tuesday of each month

California VIC Users Group
c/o Data Equipment Supply Corp.
8315 Firestone Blvd.
Downey, CA 90241
(213) 923-9361
Meetings: Second Tues. of each month

Valley Computer Club
1913 Booth Road
Ceres, CA 95307

PUG of Silicon Valley
22355 Rancho Ventura Road
Cupertino, CA 95014

Lincoln Computer Club
750 E. Yosemite
Manteca, CA 95336
John Fung, Advisor

PET on the Air
525 Crestlake Drive
San Francisco, CA 94132
Max J. Babin, Secretary

PALS (Pets Around) Livermore Society
886 South K
Livermore, CA 94550
(415) 449-1084
Every third Wednesday
7:30 p.m.
Contact: J. Johnson

SPHINX
7615 Leviston Ave.
El Cerrito, CA 94530
(415) 527-9286
Bill MacCracken

San Diego PUG
c/o D. Costarakis
3562 Union Street
(714) 235-7626
7 a.m.-4 p.m.

Walnut Creek PET Users Club
1815 Ygnacio Valley Road
Walnut Creek, CA 94596

Jurupa Wizards
8700 Galena St.
Riverside, CA 92509
781-1731
Walter J. Scott

The Commodore Connection
2301 Mission St.
Santa Cruz, CA 95060
(408) 425-8054
Bud Massey

San Fernando Valley Commodore Users Group
21208 Nashville
Chatsworth, CA 91311
(213) 709-4736
Tom Lynch
2nd Wed. 7:30

VACUUM
277 E. 10th Ave.
Chico, CA 95926
(916) 891-8085
Mike Casella
2nd Monday of month

VIC 20 Users Group
2791 McBride Ln. #121
Santa Rosa, CA
(707) 575-9836
Tyson Verse

South Bay Commodore Users Group
1402 W. 218th St.
Torrance, CA 90501
Contact: Earl Evans

Slo VIC 20/64 Computer Club
1766 9th St.
Los Osos, CA

The Diamond Bar R.O.P. Users Club
2644 Amelgado
Haciendo Hgts., CA 91745
(213) 333-2645
Don McIntosh

Commodore Interest Association
c/o Computer Data
14660 La Paz Dr.
Victorville, CA 92392
Mark Finley

Fairfield VIC 20 Club
1336 McKinley St.
Fairfield, CA 94533
(707) 427-0143
Al Brewer
1st & 3rd Tues. at 7 p.m.

Computer Barn Computer Club
319 Main St.
Suite #2
Salinas, CA 93901
757-0788
S. Mark Vanderbilt

Humboldt Commodore Group
P.O. Box 570
Arcata, CA 95521
R. Turner

Napa Valley Commodore Computer Club
c/o Liberty Computerware
2680 Jefferson St.
Napa, CA 94558
(707) 252-6281
Mick Winter
1st & 3rd Mon. of month

S.D. East County C-64 User Group
6353 Lake Apopka Place
San Diego, CA 92119
(619) 698-7814
Linda Schwartz

Commodore Users Group
4237 Pulmeria Ct.
Santa Maria, CA 93455
(805) 937-4174
Gilbert Vela

Bay Area Home Computer Asso.
1332 Pine St.
Walnut Creek, CA 94598
(415) 932-5447
Cliff Downing

Amateurs and Artesians Computing
P.O. Box 682
Cobb, CA 95426

Manteca VIC 20 Users Organization
429 N. Main St.
Manteca, CA 95336
Gene Rong

Pomona Valley Commodore Users Group
1401 W. 9th, #77
Pomona, CA 91766
(714) 620-8889
Mark Joerger
1st & 3rd Wed. of month 7 p.m.

VIC TORII-The VIC 20 Users Group
PSC #1, Box 23467
APO San Francisco, CA 96230
Wesley Clark

The Valley Computer Club
2006 Magnolia Blvd.
Burbank, CA 91506
1st Wed. 7 p.m.

The Commodore Tech. Users
of Orange Co.
P.O. Box 1497
Costa Mesa, CA 92626
(714) 731-5195
Roger Fisher

VIC 20 Software Exchange Club
10530 Sky Circle
Grass Valley, CA 95945
Daniel Upton

C-64 West Orange County
Users Group
P.O. Box 1457
Huntington Beach, CA 92647
(714) 842-4484
Philip Putman
2nd & 4th Tues. of month

Antelope Valley Commodore
Users Group
POB 4436
Lancaster, CA 93539
(805) 942-2626
James Haner
1st Saturday

Diablo Valley Commodore
Users Group
762 Ruth Dr.
Pleasant Hill, CA 94523
(415) 671-0145
Ben Braver
2nd & 4th Thurs. 7:30 p.m.

Commodore Connection
11652 Valverde Ave.
Riverside, CA 92505
(714) 689-7447
Tony Alvarez

CA. Area Commodore Terminal
Users Society
C.A.C.T.U.S.
P.O. Box 1277
Alta Loma, CA 91701
Darrell Hall

20/64
P.O. Box 18473
San Jose, CA 95158
(408) 978-0546
1st Sun. of month (6-9 p.m.)

8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Software 64
353 California Dr.
Burlingame, CA 94010
(415) 340-7115
Mario Abad

Sacramento Commodore Users Group
8120 Sundance Dr.
Orangevale, CA 95662
(916) 969-2028
Robyn Graves

Peninsula Commodore Users Group
549 Old County Rd.
San Carlos, CA 94070
(415) 593-7697
Timothy Very
2nd Thurs. of Month

San Francisco Commodore
Users Group
278-27th Ave. #103
San Francisco, CA 94121
(415) 387-0225
Roger Tierce

South Bay Commodore 64
Users Group
P.O. Box 3193
San Ysidro, CA 95073

Commodore 64 West Computer Club
2917 Colorado Ave.
Santa Monica, CA 90404
(213) 828-9308
Don Campbell

Sixty Fourum
4413 E. Iowa
Fresno, CA 93702
(209) 252-0392
Deb Christensen

C-64/VIC 20 Users Group
Pasadena City College
Cicadian Room
Pasadena, CA
(714) 593-4880
Chuck Cypher
7 p.m. 1st & 3rd Thursdays

Marin Commodore Computer Club
620 Del Ganado Rd.
San Rafael, CA
(415) 479-0426
2nd Wed. of month 7:30 p.m.

COLORADO

VICKIMPET Users Group
4 Waring Lane, Greenwood
Village
Littleton, CO 80121
Contact: Louis Roehrs

Colorado Commodore
Computer Club
2187 S. Golden Ct.
Denver, CO 80227
986-0577
Jack Moss
Meet: 2nd Wed.

Commodore Users Group
Box 377
Aspen, CO 81612
(303) 925-5604
Ray Brooks
1st Monday in the evenings

CONNECTICUT

John F. Garbarino
Skiff Lane Masons Island
Mystic, CT 06355
(203) 536-9789

Commodore User Club
Wethersfield High School
411 Wolcott Hill Road
Wethersfield, CT 06109
Contact: Daniel G. Spaneas

VIC Users Club
c/o Edward Barszczewski
22 Tunxis Road
West Hartford, CT 06107

New London County
Commodore Club
Doolittle Road
Preston, CT 06360
Contact: Dr. Walter Doolittle

The Commodore East Users Group
165 B S. Bigelow Rd.
Hampton, CT 06247
(203) 455-0108

DISTRICT OF COLUMBIA

USO Computer Club
USO Outreach Center
207 Beyer Rd., SW
Washington, DC 20332
Steven Guenther

FLORIDA

Jacksonville Area
PET Society
401 Monument Road, #177
Jacksonville, FL 32211

Richard Prestien
6278 SW 14th Street
Miami, FL 33144

South Florida
PET Users Group
Dave Young
7170 S.W. 11th
West Hollywood, FL 33023
(305) 987-6982

PETs and Friends
129 NE 44 St.
Miami, FL 33137
Richard Plumer

Sun Coast VICs
P.O. Box 1042
Indian Rocks Beach, FL
33535
Mark Weddell

Bay Commodore Users
Group
c/o Gulf Coast Computer
Exchange
241 N. Tyndall Pkwy.
P.O. Box 6215
Panama City, FL 32401
(904) 785-6441
Richard Scofield

Gainesville Commodore
Users Club
3604-20A SW 31st Dr.
Gainesville, FL 32608
Louis Wallace

Brandon Users Group
108 Anglewood Dr.
Brandon, FL 33511
(813) 685-5138
Paul Daugherty

Brandon Commodore Users Group
414 E. Lumsden Rd.
Brandon, FL 33511

Gainesville Commodore
Users Group
Santa Fe Community College
Gainesville, FL 32602
James E. Birdsell

Commodore Computer Club
P.O. Box 21138
St. Petersburg, FL 33742
(813) 522-2547
Chuck Fechko

Commodore Users Group
545 E. Park Ave.
Apt. #2
Tallahassee, FL 32301
(904) 224-6286
Jim Neill

The Commodore Connection
P.O. Box 6684
West Palm Beach, FL 33405

El Shift OH
P.O. Box 548
Cocoa, FL 32922
Mike Schnoke
Sat. mornings/every 4 to 6 weeks

Miami 20/64
12911 S.W. 49th St.
Miami, FL 33175
(305) 226-1185

Tampa Bay Commodore
Computer Club
10208 N. 30th St.
Tampa, FL 33612
(813) 977-0877

Commodore Computer Club
P.O. Box 9726
Jacksonville, FL 32208
(904) 764-5457
David Phillips
2nd & 4th Tues. of Month

VIC/64 Heartland Users Group
1220 Bartow Rd. #23
Lakeland, FL 33801
(813) 666-2132
Tom Keough
4th Wed. of Month at PRC

64 Educators Users Group South
FDLRS-South
9220 S.W. 52nd Terrace
Miami, FL 33165
(305) 274-3501
Dr. Eydie Sloane

64 Educators Users Group North
16330 N.E. 2nd Ave.
North Miami Beach, FL 33162
(305) 944-5548
Robert Figueroa

Suncoast 64S
c/o Little Professor Book Center
2395 U.S. 19 North
Palm Harbor, FL 33563
(813) 785-1036
Curtis Miller

Lakeland VIC 20 Users Group
2450 Shady Acres Dr.
Mulberry, FL 33860

Broward Commodore Users Group
13 Spinning Wheel Lane
Tamarac, FL 33319
(305) 726-4390
Lewis Horn

GEORGIA

VIC Educators Users Group
Cherokee County Schools
110 Academy St.
Canton, GA 30114
Dr. Al Evans

Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

VIC-tims
P.O. Box 467052
Atlanta, GA 30346
(404) 922-7088
Eric Ellison

Golden Isles Commodore
Users Club
Bldg. 68, FLETC
Glynco, GA 31524
Richard L. Young

user groups

Commodore Club of Augusta
1011 River Ridge Rd.
Apt. #14-A
Augusta, GA 30909
David Dumas

Dataswapper Users Group
1794 Alabama Ave.
Albany, GA 31705
(912) 436-5596
David Via

HAWAII

Commodore Users Group of Honolulu
c/o PSH
824 Bannister St.
Honolulu, HI
(808) 848-2088
3rd Fri. every month

20/64 Hawaii
P.O. Box 966
Kailua, HI 96734
Wes Goodpaster

Commodore Users Group of Honolulu
1626 Wilder #701
Honolulu, HI 96822
(808) 848-2088
Jay Calvin (808) 944-9380

IDAHO

GHS Computer Club
c/o Grangeville High School
910 S. D St.
Grangeville, ID 83530
Don Kissinger

S.R.H.S. Computer Club
c/o Salmon River H.S.
Riggins, ID 83549
Barney Foster

Commodore Users
548 E. Center
Pocatello, ID 83201
(208) 233-0670
Leroy Jones

Eagle Rock Commodore Users Group
900 S. Emerson
Idaho Falls, ID 83401
Nancy J. Picker

64 Bug (Boise Users Group)
P.O. Box 276
Boise, ID 83701
(208) 344-6302
John Rosecrans

ILLINOIS

Shelly Wernikoff
2731 N. Milwaukee
Avenue
Chicago, IL 60647

VIC 20/64 Users Support
Group
c/o David R. Tarvin
114 S. Clark Street
Pana, IL 62557
(217) 562-4568

Central Illinois PET User
Group
635 Maple
Mt. Zion, IL 62549
(217) 864-5320
Contact: Jim Oldfield

ASM/TED User Group
200 S. Century
Rantoul, IL 61866
(217) 893-4577
Contact: Brant Anderson

PET VIC Club (PVC)
40 S. Lincoln
Mundelein, IL 60060
Contact: Paul Schmidt,
President

Rockford Area PET Users
Group
1608 Benton Street
Rockford, IL 61107

Commodore Users Club
1707 East Main St.
Olney, IL 62450
Contact: David E. Lawless

VIC Chicago Club
3822 N. Bell Ave.
Chicago, IL 60618
John L. Rosengarten

Chicago Commodore 64
Users & Exchange Group
P.O. Box 14233
Chicago, IL 60614
Jim Robinson

Fox Valley PET Users
Group
833 Willow St.
Lake in the Hills, IL 60102
(312) 658-7321
Art DeKneef

The Commodore 64 Users
Group
P.O. Box 572
Glen Ellyn, IL 60137
(312) 790-4320
Gus Pagnotta

RAP 64/VIC Regional
Assoc. of Programmers
10721 S. Lamon
Oak Lawn, IL 60453
Bob Hughes

The Kankakee Hackers
RR #1, Box 279
St. Anne, IL 60964
(815) 933-4407
Rich Westerman

WIPUG
Rt. 5, Box 75
Quincy, IL 62301
(217) 656-3671
Edward Mills

Papug-Peoria Area Pet Users Group
6 Apple Tree Lane
East Peoria, IL 61611
(309) 673-6635
Max Taylor
2nd Fri. of Month

McHenry County Commodore Club
4900 S. Route 31
Crystal Lake, IL 60014
(815) 455-3942
John Katkus
2nd Sat. of month 9 to 12 a.m.

INDIANA

PET/64 Users
10136 E. 96th St.
Indianapolis, IN 46256
(317) 842-6353
Jerry Brinson

Cardinal Sales
6225 Coffman Road
Indianapolis, IN 46268
(317) 298-9650
Contact: Carol Wheeler

CHUG (Commodore
Hardware Users Group)
12104 Meadow Lane
Oaklandon, IN 46236
Contact: Ted Powell

VIC Indy Club
P.O. Box 11543
Indianapolis, IN 46201
(317) 898-8023
Ken Ralston

Northern Indiana
Commodore Enthusiasts
927 S. 26th St.
South Bend, IN 46615
Eric R. Bean

Commodore Users Group
1020 Michigan Ave.
Logansport, IN 46947
(219) 722-5205
Mark Bender

Computer Workshop VIC 20/64 Club
282 S. 600 W.
Hebron, IN 46341
(219) 988-4535
Mary O'Bringer

The National Science Clubs
of America
Commodore Users Division
7704 Taft St.
Merrillville, IN 46410
Brian Lepley or Tom Vlasic

East Central Indiana VIC User Group
Rural Route #2
Portland, IN 47371
Stephen Erwin

National VIC 20 Program Exchange
102 Hickory Court
Portland, IN 47371
(219) 726-4202
Stephen Erwin

Commodore Computer Club
3814 Terra Trace
Evansville, IN 47711
(812) 477-0739
John Patrick, President

Commodore 64 Users Group
912 South Brown Ave.
Terre Haute, IN 47803
(812) 234-5099
Dennis Graham

Seymour Peekers
c/o D&L Camera Shop
108 N. Chestnut
Seymour, IN 47274
Dennis Peters

IOWA

Commodore User Group
114 8th St.
Ames, IA 50010

Quad City Commodore Club
1721 Grant St.
Bettendorf, IA 52722
(319) 355-2641
John Yigas

Siouxland Commodore Club
2700 Sheridan St.
Sioux City, IA 51104
(712) 258-7903
Gary Johnson
1st & 3rd Monday of month

421 W. 6th St.
Waterloo, IA 50702
(319) 232-1062
Frederick Volker

Commodore Computer Users
Group of Iowa
Box 3140
Des Moines, IA 50316
(515) 263-0963 or (515) 287-1378
Laura Miller

Commo-Hawk Commodore
Users Group
P.O. Box 2724
Cedar Rapids, IA 52406
Vern Rotert

KANSAS

Wichita Area PET
Users Group
2231 Bullinger
Wichita, KS 67204
(316) 838-0518
Contact: Mel Zandler

Kansas Commodore
Computer Club
101 S. Burch
Olathe, KS 66061
Contact: Paul B. Howard

Commodore Users Group
6050 S. 183 St. West
Viola, KS 67149
Walter Lounsbery

Walnut Valley Commodore
User Group
1003 S. 2nd St.
Arkansas City, KS 67005
Bob Morris

KENTUCKY

VIC Connection
1010 S. Elm
Henderson, KY 42420
Jim Kemp

Louisville Users of Commodore KY.
(LUCKY)
P.O. Box 22244
Louisville, KY 40222
(502) 425-2847
2nd Tues. of Month

The Bowling Green Commodore
Users Group
Route 11, Creekside Apt. #6
Bowling Green, KY 42101
(502) 781-9098
Alex Fitzpatrick

LOUISIANA

Franklin Parish Computer
Club
#3 Fair Ave.
Winnisboro, LA 71295
James D. Mays, Sr.

NOVA
917 Gordon St.
New Orleans, LA 70117
(504) 948-7643
Kenneth McGruder, Sr.

VIC 20 Users Group
5064 Bowdon St.
Marrero, LA 70072
(504) 341-5305
Wayne D. Lowery, R.N.

64-Club News
5551 Corporate Blvd.
Suite 3L

Baton Rouge, LA 70808
(504) 766-7408
Tom Parsons
3rd Tues. of month at CWA

Commodore Users Group of Oachita
P.O. Box 175
Swaric, LA 71281
(318) 343-8044
Beckie Walker

Ark-La-Tex Commodore 64 Club
198 India Dr.
Shreveport, LA 71115
(318) 797-9702
Pete Whaley

Commodore 64 Users Group
P.O. Box 1422
Baton Rouge, LA 70821
Richard Hood
3rd Tues. of month

MAINE

COM-VICS (Commodore/VIC Users Group)
RFD #1, Box 2086
Hebron, ME 04238
(207) 966-3641
Paul Lodge
1st Wed. & 3rd Thurs.

Your Commodore Users Group
Box 611
Westbrook, ME 04092
(207) 854-4579
Mike Procise

MARYLAND

Assoc. of Personal Computer Users
5014 Rodman Road
Bethesda, MD 20016

Blue TUSK
700 East Joppa Road
Baltimore, MD 21204
Contact: Jim Hauff

House of Commodore
8835 Satyr Hill Road
Baltimore, MD 21234
Contact: Ernest J. Fischer

Long Lines Computer Club
323 N. Charles St., Rm. 201
Baltimore, MD 21201
Gene Moff

VIC & 64 Users Group
The Boyds Connection
21000 Clarksburg Rd.
Boyds, MD 20841
(301) 428-3174
Tom DeReggi

Rockville VIC/64 Users Group
P.O. Box 8805
Rockville, MD 20856
(301) 231-7823
Tom Pounds

The Compucats' Commodore Computer Club
680 W. Bel Air Ave.
Aberdeen, MD 21001
(301) 272-0472
Betty Jane Schueler

Westinghouse BWI Commodore User Group
Attn: L. Barron Mail Stop 5156
P.O. Box 1693
Baltimore, MD 21203

HUG (Hagerstown Users Group)
23 Coventry Lane
Hagerstown, MD 21740
(301) 797-9728
Joseph Rutkowski

The Montgomery Ct. Commodore Computer Soc.
P.O. Box 6444
Silver Springs, MD 20906
(301) 946-1564
Meryle Pounds

Commodore Users Group of Annapolis
P.O. Box 9726
Arnold, MD 21012
(301) 974-4548
The Software Co.

MASSACHUSETTS

Eastern Massachusetts VIC Users Group
c/o Frank Ordway
7 Flagg Road
Marlboro, MA 02173

VIC Users Group
c/o Ilene Hoffman-Sholar
193 Garden St.
Needham, MA 02192

Commodore Users Club
Stoughton High School
Stoughton, MA 02072
Contact: Mike Lennon

Berkshire PET Lovers
CBM Users Group
Taconic High
Pittsfield, MA 01201

The Boston Computer Society
Three Center Plaza
Boston, MA 02108
(617) 367-8080
Mary E. McCann

Masspet Commodore Users Group
P.O. Box 307
East Taunton, MA 02718
David Rogers

Raytheon Commodore Users Group
Raytheon Company
Hartwell Rd. GRA-6
Bedford, MA 01730
John Rudy

Commodore 64 Users Group of The Berkshires
184 Highland Ave.
Pittsfield, MA 01201
Ed Rucinski

VIC Interface Club
48 Van Cliff Ave.
Brockton, MA 02401
Bernie Robichaud

Cape Cod 64 Users Group
358 Forrest Rd.
S. Yarmouth, MA 02664
1 (800) 225-7136
Jim Close
(In MA. call) 1 (800) 352-7787

The Cursor Club
442 Mulpuf Rd.
Lunenburg, MA 01462
(617) 582-0529
John

Pioneer Valley VIC/64 Club
34 Bates St.
Westfield, MA 01085
(413) 562-1027
Marvin Yale
3rd Thurs. of month

MICHIGAN

David Liem
14361 Warwick Street
Detroit, MI 48223

VIC Users Club
University of Michigan
School of Public Health
Ann Arbor, MI 48109
Contact: John Gannon

Commodore User Club
32303 Columbus Drive
Warren, MI 48093
Contact: Robert Steinbrecher

Commodore Users Group
c/o Family Computer
3947 W. 12 Mile Rd.
Berkley, MI 48072

VIC for Business
6027 Orchard Ct.
Lansing, MI 48910
Mike Marotta

South Computer Club
South Jr. High School
45201 Owen
Belleville, MI 48111
Ronald Ruppert

Commodore Users Group
c/o Eaton Rapids Medical Clinic
101 Spicerville Hwy.
Eaton Rapids, MI 48827
Albert Meinke III, M.D.

South East Michigan Pet Users Group
Box 214
Farmington, MI 48024
Norm Eisenberg

Commodore Computer Club
4106 Eastman Rd.
Midland, MI 48640
(517) 835-5130
John Walley
9:30 p.m. Sept/May

VIC, 64, PET Users Group
8439 Arlis Rd.
Union Lake, MI 48085
363-8539
Bert Searing

COMP
486 Michigan Ave.
Marysville, MI 48040
(313) 364-6804
M. Gauthier

Ann Arbor Commodore Users Group
Ann Arbor, MI 48103
(313) 994-4751
Art Shaw
3rd Tues. 7:30-10:00

DAB Computer Club
P.O. Box 542
Watervliet, MI 49098
(616) 463-5457
Dennis Burlingham

West Michigan Commodores
c/o R. Taber
1952 Cleveland Ave., S.W.
Wyoming, MI 49509
(616) 458-9724
Gene Traas

Debug
P.O. Box 196
Berrien Springs, MI 49103
(616) 471-1882
Herbert Edward
Last Thursday of Month

Jackson Commodore Computer Club
201 S. Grinnell St.
Jackson, MI 49203
Alfred Bruey
Last Thur. of Month 7:30 p.m.

SMCUG
1002 Pfau St.
Mankato, MI 56001
(507) 625-6942
Dean Otto

SEM 64
25015 Five Mile #3
Redford, MI 48239
(313) 537-4163
Gary Groeller

C.A.T.O.
17606 Valade
Riverview, MI 48192
Dean Tidwell

MINNESOTA

MUPET (Minnesota Users of PET)
P.O. Box 179
Annandale, MN 55302
c/o Jon T. Minerich

Twin Cities Commodore Computer Club
6623 Ives Lane
Maple Grove, MN 55369
(612) 424-2425
Contact: Rollie Schmidt

Brainerd Area Commodore Users Group
1219 S.E. 11th St.
Brainerd, MN 56401
(218) 829-0805
Norm Saavedra
1st Thurs. 6 p.m. & 3rd Sat. 10 a.m.

MISSISSIPPI

Commodore Biloxi User Group (ComBUG)
Universal Computer Services
3002 Hwy. 90 East
Ocean Springs, MS 39564
(601) 875-1173
John Lassen

MISSOURI

KCPUG
(Commodore User Group of Kansas City)
P.O. Box 36492
Kansas City, MO 64111
(816) 252-7628
Salvadore

Commodore User Group of St. Louis
Box 6653
St. Louis, MO 63125-0653
Dan Weidman, New Members
1541 Swallowtail Dr.
St. Louis, MO

VIC INFONET
P.O. Box 1069
Branson, MO 65616
(417) 334-6099
Jory Sherman

user groups

Worth County PET Users Group
Grant City, MO
(816) 564-3551
David Hardy

Mid-Missouri Commodore Club
1804 Vandiver Dr.
Columbia, MO 65201
(314) 474-4511
Phil Bishop

Joplin Commodore Computers Users Group
422 S. Florida Ave.
Joplin, MO 64801
R. D. Connely

MONTANA

Powder River Computer Club
Powder River County High School
Broadus, MT 59317
Contact: Jim Sampson

Commodore User Club
1109 West Broadway
Butte, MT 59701
Contact: Mike McCarthy

NEBRASKA

Greater Omaha Commodore 64 Users Group
2932 Leawood Dr.
Omaha, NE 68123
(402) 292-2753
Bob Quisenberry

NEVADA

Las Vegas PET Users
Suite 5-315
5130 E. Charleston Blvd.
Las Vegas, NV 89122
Gerald Hasty

Compu Club 64
4220 S. Maryland Parkway
Bldg. B—Suite 403
Las Vegas, NV 89109
(702) 369-7354
Cindy Springfield

NEW JERSEY

Commodore Friendly User Group
49 Hershey Rd.
Wayne, NJ 07470
(201) 696-8043
Rich Pinto/Colin Campbell

Somerset Users Club
49 Marcy Street
Somerset, NJ 08873
Contact: Robert Holzer

Educators Advisory
P.O. Box 186
Medford, NJ 08055
(609) 953-1200
John Handfield

VIC-TIMES
46 Wayne Street
Edison, NJ 08817
Thomas R. Molnar

VIC 20 User Group
67 Distler Ave.
W. Caldwell, NJ 07006
(201) 284-2281
G. M. Amin

VIC Software Development Club
77 Fomalhaut Ave.
Sewell, NJ 08080
H. P. Rosenberg

ACGNJ PET/VIC/CBM User Group
30 Riverview Terr.
Belle Mead, NJ 08502
(201) 359-3862
J. M. Pylka

South Jersey Commodore Users Club
46-B Monroe Path
Maple Shade, NJ 08052
(609) 667-9758
Mark Orthner

Parsippany Computer Group
51 Ferncliff Rd.
Morris Plains, NJ 07950
(201) 267-5231
Bob Searing

Information 64
16 W. Ridgewood Ave.
Ridgewood, NJ 07450
(201) 447-4432
Dave Garaffa

NEW HAMPSHIRE

Northern New England Computer Society
P.O. Box 69
Berlin, NH 03570

TBH VIC-NICs
P.O. Box 981
Salem, NH 03079

C-64 U.S.E.R.S. User Software Exchange Pro
P.O. Box 4022
Rochester, NH 03867
Paul Kyle

NEW MEXICO

Commodore Users Group
6212 Karlson, NE
Albuquerque, NM 87113
(505) 821-5812
Danny Byrne

NEW YORK

Capital District 64/VIC 20 Users Group
363 Hamilton St.
Albany, NY 12210
(518) 436-1190
Bill Pizer

Long Island PET Society
Ralph Bressler
Harborfields HS
Taylor Avenue
Greenlawn, NY 11740

PET User Club of Westchester
P.O. Box 1280
White Plains, NY 10602
Contact: Ben Meyer

LIVE (Long Island VIC Enthusiasts)
17 Picadilly Road
Great Neck, NY 11023
Contact: Arnold Friedman

Commodore Masters
25 Croton Ave.
Staten Island, NY 10301
Contact: Stephen Farkouh

VIC Users Club
76 Radford St.
Staten Island, NY 10314
Contact: Michael Frantz

West Chester County VIC Users Group
P.O. Box 146
Pelham, NY 10552
Joe Brown

SPUG
4782 Boston Post Rd.
Pelham, NY 10803
Paul Skipski

VIC 20 User Club
151-28 22nd Ave.
Whitestone, NY 11357
Jean F. Coppola

VIC 20 User Club
339 Park Ave.
Babylon, NY 11702
(516) 669-9126
Gary Overman

VIC User Group
1250 Ocean Ave.
Brooklyn, NY 11230
(212) 859-3030
Dr. Levitt

L&M Computer Club
VIC 20 & 64
4 Clinton St.
Tully, NY 13159
(315) 696-8904
Dick Mickelson

Commodore Users Group
1 Corwin Pl.
Lake Katrine, NY 12449
J. Richard Wright

VIC 20/Commodore 64 Users Group
31 Maple Dr.
Lindenhurst, NY 11757
(516) 957-1512
Pete Lobol

VIC Information Exchange Club
336 W. 23 St.
Deer Park, NY 11729
Tom Schlegel
SASE & phone please

New York Commodore Users Group
380 Riverside Dr., 7Q
New York, NY 10025
(212) 566-6250
Ben Tunkelang

Hudson Valley Commodore Club
1 Manor Dr.
Woodstock, NY 12498
F.S. Goh
1st Wednesday of month

LIVICS (Long Island VIC Society)
20 Spyglass Lane
East Setauket, NY 11733
(516) 751-7844
Lawrence Stefani

VIC Users Group
c/o Stoney Brook Learning Center
1424 Stoney Brook Rd.
Stoney Brook, NY 11790
(516) 751-1719
Robert Wurtzel

Poughkeepsie VIC User Group
2 Brooklands Farm Rd.
Poughkeepsie, NY 12601
(914) 462-4518
Joe Steinman

VIC 20 User Group
Paper Service Division
Kodak Park
Rochester, NY 14617
David Upham, Sr.

Manhattan 64
c/o Steve Lazarowitz
1440 Freeport Loop
Brooklyn, NY 11239
(212) 647-4266
Larry Thompson

Adirondack Commodore 64 Users Group
205 Woodlawn Ave.
Saratoga Springs, NY
(518) 584-8960
Paul Klompas

Rockland County Commodore Users Group
P.O. Box 573
Nanuet, NY 10965
Ross Garber

New York 64 Users Group
222 Thompson St.
New York, NY 10012
(212) 673-7241
Bruce Cohen

Finger Lakes Commodore Users Group
c/o Rose City Computer Associates
229 West Union St.
Newark, NY 14513
(315) 331-1185

The Commodore Users Group Rochester
78 Hardison Rd.
Rochester, NY 14617
(716) 544-5251
Tom Werenski
Phone Evenings between 7-10

Commodore Computer Club
Publications Dept.,
Grumman Aerospace
1111 Stewart Ave.
Bethpage, NY 11714
(516) 575-9558
Neil Threulsen

Hello, Central!
76-12 35th Ave.
Jackson Heights, NY 11372
Jared Sherman

VIC 20/64 Users Group
NYU Waverly Place
New York, NY 10003
(212) 358-5155
Lawrence Schulman

SCUG (Schenectady Commodore Users Group)
c/o the Video Connection
Canal Square
Schenectady, NY 12305
Timothy Davis
1st Mon. of Month

Commodore 64 Users Group
S.U.N.Y. at Oswego
Dept. of Industrial Arts
Oswego, NY 13126
John R. Boronkay

NORTH CAROLINA

Amateur Radio PET Users Group
P.O. Box 30694
Raleigh, NC 27622
Contact: Hank Roth

VIC Users Club
c/o David C. Fonenberry
Route 3, Box 351
Lincolnton, NC 28092

Microcomputer Users Club
Box 17142 Bethabara Sta.

Winston-Salem, NC 27116
Joel D. Brown

VIC Users Club
Rt. 11, Box 686
Hickory, NC 28601
Tim Gromlovits

Raleigh VIC 20/64 Users Group
410-D Delta Court
Cary, NC 27511
(919) 469-3862
Larry Diener

Down East Commodores
302 Belltown Rd.
Havelock, NC 28532
(919) 447-4536
Bruce Thedin

Tryon Commodore 64 Club
P.O. Box 1016
Tryon, NC 28782
(704) 859-6340
Robin Michael
1st Mon. of month at 7 p.m.

OHIO

Dayton Area PET
User Group
933 Livingston Drive
Xenia, OH 45385
B. Worby, President
(513) 848-2065
J. Watson, Secretary
(513) 372-2052

Central Ohio PET
Users Group
107 S. Westmoor Avenue
Columbus, OH 43204
(614) 274-0304
Contact: Philip H. Lynch

Commodore Computer Club
of Toledo
734 Donna Drive
Temperance, MI 48182
Gerald Carter

Chillicothe Commodore
Users Group
P.O. Box 211
Chillicothe, OH 45601
William A. Chaney

Licking County 64 Users Group
323 Schuler St.
Newark, OH 43055
(614) 345-1327

11433 Pearl Rd.
Strongsville, OH 44136
Paul M. Warner

C.P.U. Connection
P.O. Box 42032
Brook Park, OH 44142
Danni Hudak

SE Cleveland Commodore Crazies
18813 Harlan Dr.
Maple Heights, OH 44137
(216) 581-3099
Carl Skala

Commodore Users of
Blue Chip (Cincinnati)
816 Beecher St.
Cincinnati, OH 45206
(513) 961-6582
Ted Stalets

Southwestern Ohio Commodore
Users Group
P.O. Box 399117
Cincinnati, OH 45239
2nd Wed. of month at 7 p.m.

S.W.O.C.U.G. (SW. Ohio
Commodore Users Gp.)
8401 Wicklow Ave.
Cincinnati, OH 45236
Joe Beresford

OKLAHOMA

Southwest Oklahoma
Computer Club
c/o Commodore Chapter
P.O. Box 6646
Lawton, OK 73504
1:30 at Lawton City Library

Tulsa Area Commodore Users Group
Tulsa Computer Society
P.O. Box 15238
Tulsa, OK 74112
Annette Hinshaw

Commodore Oklahoma Users Club
4000 NW 14th St.
Oklahoma City, OK 73107
(405) 943-1370
Stanley B. Dow

Commodore Users
Box 268
Oklahoma City, OK 73101
Monte Maker, President

Commodore Users of Norman
209 Brookwood
Noble, OK 73068
Matt Hager

Commodore Users Group
Muskogee Computer Society
202 S. 12th St.
Muskogee, OK 74401
Steve Ford

OREGON

NW PET Users Group
John F. Jones
2134 N.E. 45th Avenue
Portland, OR 97213

U.S. Commodore Users Group
P.O. Box 2310
Roseburg, OR 97470
(503) 672-7591
Richard Tsukiji

Southern Oregon VIC/64
Users Group
3600 Madrona Lane
Medford, OR 97501
(503) 779-7631
James Powell

PENNSYLVANIA

PET User Group
Gene Beals
P.O. Box 371
Montgomeryville, PA 18936

Penn Conference Computer Club
c/o Penn Conference of SDA
720 Museum Road
Reading, PA 19611
Contact: Dan R. Knepp

PACS Commodore Users Group
LaSalle College
20th & Olney Ave.
Philadelphia, PA 19141
(215) 951-1258
Stephen Longo

Glen Schwartz
807 Avon
Philadelphia, PA 19116

Gene Planchak
4820 Anne Lane
Sharpsville, PA 15150
(412) 962-9682

PPG (Pittsburgh PET Group)
c/o Joel A. Casar, DMD
2015 Garrick Drive
Pittsburgh, PA 15235
(412) 371-2882

Westmoreland Commodore
Users Club
c/o DJ & Son Electronics
Colonial Plaza
Latrobe, PA 15650
Jim Mathers

Commodore Users Club
3021 Ben Venue Dr.
Greensburg, PA 15601
(412) 836-2224
Jim Mathers

VIC 20 Programmers, Inc.
c/o Watson Woods
115 Old Spring Rd.
Coatesville, PA 19320
Robert Gougher

G.R.C. User Club
300 Whitten Hollow Rd.
New Kensington, PA 15068
Bill Bolt

NADC Commodore Users Club
248 Oakdale Ave.
Horsham, PA 19044
Norman McCrary

CACC (Capitol Area Commodore
Club)
134 College Hill Rd.
Enola, PA 17025
(717) 732-2123
Lewis Buttery
Union Deposit Mall at 7 p.m.

G/C Computer Owners Group
c/o Gilbert Associates, Inc.
P.O. Box 1498
Reading, PA 19607
Extension 6472
Jo Lambert (215) 775-2600

Boeing Employees Personal
Computer Club
The Boeing Vertol Co.
P.O. Box 16858
Philadelphia, PA 19142
(215) 522-2257
Jim McLaughlin

South Central PA Commodore Club
2109 Cedar Run Dr.
Camp Hill, PA 17011
(717) 763-4219
David Persing

Main Line Commodore Users
Group (MLCUG)
c/o Main Line Computer Center
1046 General Allen Lane
West Chester, PA 19380
(215) 388-1581
Emil Volcheck

Commodore Users Group
781 Dick Ave.
Warminster, PA 18974
Matt Matulaitis

The Commodore Users Club
of S.E. Pittsburgh
c/o Groves Appliance & TV
2407 Pennsylvania Ave.
West Mifflin, PA 15122
Charles Groves

Compstars
130 Blue Teal Circle
Audubon, PA 19403
Mike Norm
Meet at Audio Video Junct.

Scranton Commodore Users Group
P.O. Box 211
Clarks Summit, PA 18411

Clifton Heights Users Group
P.O. Box 235
Clifton Heights, PA 19018

Oxford Circle 64 Users Group
Frankford Cong. Un. Church of Christ
Oxford Ave. & Pratt St.
Philadelphia, PA 19124
(215) 743-8999
Roger Nazeley (215) 535-9021
4th Wed. of Month

VIC Software Development Club
440 W. Sedgwick
Apt. A-1
Philadelphia, PA 19119
(215) 844-4328
Tracy Lee Thomas

Bits & Bytes'
1015 Dale Rd.
Secane, PA 19018
(215) 544-5875
Dave Boodey

CACCC-Centre Area Commodore
Computer Club
214 Computer Building
University Park, PA 16802
(814) 237-5912
Bill Hillner

PUERTO RICO

CUG of Puerto Rico
RFD #1, Box 13
San Juan, PR 00914
Ken Burch

VIC 20 User Group
655 Hernandez St.
Miramar, PR 00907
Robert Morales, Jr.

RHODE ISLAND

Irving B. Silverman, CPA
160 Taunton Ave.
E. Providence, RI 02914
Contact: Michelle Chavanne

Newport VIC/64 Users
10 Maitland Ct.
Newport, RI 02840
(401) 849-2684
Dr. Matt McConeghy

The VIC 20 Users Club
Warwick, RI 02886
Tom Davey

Commodore Users Group
c/o Data-Co.
978 Tiogue Ave.
Coventry, RI 02816
(401) 828-7385
Victor Moffett

SOUTH CAROLINA

Beaufort Technical College
100 S. Ribaut Rd.
Beaufort, SC 29902
Dean of Instruction

Computer Users Society
of Greenville (CUS)
Horizon Records-Home Computers
347 S. Pleasantburg Dr.

user groups

Greenville, SC 29607
(803) 235-7922
Bo Jeanes

Commodore Computer Club
of Columbia
318 Quincannon Dr.
Columbia, SC 29210
Buster White Sect./Treas.

Spartanburg Commodore
Users Group
803 Lucerne Dr.
Spartanburg, SC 29302
(803) 582-5897
James Pasley

SOUTH DAKOTA

PET User Group
515 South Duff
Mitchell, SD 57301
(605) 996-8277
Contact: Jim Dallas

VIC/64 Users Club
608 West 5th
Pierre, SD 57501
(605) 224-4863
Larry Lundeen

TENNESSEE

River City Computer
Hobbyists
Memphis, TN
1st Mon. at Main Library

Nashville Commodore Users Group
P.O. Box 121282
Nashville, TN 37212
(615) 331-5408
Dave Rushing
3rd Thurs. at Cumberland Mus

Commodore User Club
Metro Computer Center
1800 Dayton Blvd.
Chattanooga, TN 37405
Mondays 7:30 pm

Metro-Knoxville Commodore
Users Club
7405 Oxmoor Rd., Rt. #20
Knoxville, TN 37921
(615) 938-3773
Ed Pritchard

Memphis Commodore Users Group
2476 Redvers Ave.
Memphis, TN 38127
(901) 358-5823
Harry Ewart

TEXAS

PET Users
2001 Bryan Tower
Suite 3800
Dallas, TX 75201

Larry Williams
P.O. Box 652
San Antonio, TX 78293

PET User Group
John Bowen
Texas A & M
Microcomputer Club
Texas A & M, TX

CHUG (Commodore Houston
Users Group)
8738 Wildforest
Houston, TX 77088
(713) 999-3650
Contact: John Walker

Commodore Users Group
5326 Cameron Rd.
Austin, TX 78723
(512) 459-1220
Dr. Jerry D. Frazee

VIC Users Group
3817 64th Dr.
Lubbock, TX 79413

64 Users Group
2421 Midnight Circle
Plano, TX 75075
S.G. Grodin

Savid Computer Club
312 West Alabama
Suite 2
Houston, TX 77006
Davi Jordan, Chairman

Gulf Coast Commodore
Users Group
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577
Lawrence Hernandez

Mid-Cities Commodore Club
413 Chisolm Trail
Hurst, TX 76053
Garry Wordelman

Mid-Cities Commodore Club
413 Chisolm Trail
Hurst, TX 76053
Bruce Nelson

Interface Computer Club
814 North Sabinas
San Antonio, TX 78207
M. E. Garza, President

Gulf Coast Commodore
Users Group
P.O. Box 128
Corpus Christi, TX 78403
(512) 887-4577
Lawrence Hernandez

ICUG (Irving Commodore
Users Group)
3237 Northgate #1289
Irving, TX 75062
(214) 252-7017
Robert Hayes

Commodore Computer Club (C3)
c/o Lamar Full Gospel Assembly
1200 S. Sumner
Pampa, TX 79065
(806) 665-3444
Randy Mills
Every other Thurs. 7 p.m.

UTAH

Utah PUG
Jack Fleck
2236 Washington Blvd.
Ogden, UT 84401

The Commodore Users
Club
742 Taylor Avenue
Ogden, UT 84404
Contact: Todd Woods Kap,
President
David J. Shreeve,
Vice President

The VIClic
799 Ponderosa Drive
Sandy, UT 84070
Contact: Steve Graham

VIC 20 Users
324 N. 300 W.
Smithfield, UT 84335
Dave DeCorso

Northern Utah VIC & 64
Users Group
P.O. Box 533
Garland, UT 84312
David Sanders

The Commodore Users Group
652 West 700 North
Clearfield, UT 84015
(801) 776-3950
Rodney Keller, Richard Brenchly

Mountain Computer Society
P.O. Box 1154
Sandy, UT 84091
Dave Tigner

VIRGINIA

Northern VA PET Users
Bob Karpen
2045 Eakins Court
Reston, VA 22091
(703) 860-9116

VIC Users Group
Rt. 2, Box 180
Lynchburg, VA 24501
Contact: Dick Rossignol

VIC Users Group
c/o Donnie L. Thompson
1502 Harvard Rd.
Richmond, VA 23226

Dale City Commodore
User Group
P.O. Box 2004
Dale City, VA 22193
(703) 680-2270
James Hogler

Tidewater Commodore
Users Group
4917 Westgrove Rd.
Virginia Beach, VA 23455
Fred Monson

Fredericksburg Area
Computer Enthusiasts
P.O. Box 324
Locust Grove, VA 22508
(703) 972-7195
Michael Parker

VIC 20 Victims
4301 Columbia Pike #410
Arlington, VA 22204
(703) 920-0513
Mike Spengel

Peninsula Commodore 64
Users Group
124 Burnham Place
Newport News, VA 23606
(804) 595-7315
Richard G. Wilmoth

Norfolk Users Group
1030 West 43rd St. B-4
Norfolk, VA 23508
489-8292
Larry Pearson

NASA VIC 20 User Group
713 York Warwick Dr.
Yorktown, VA 23692
Harris Hamilton

135 Beverley Rd.
Danville, VA 24541
David Gray

R.A.C.E. Commodore
Users Group
4726 Horseman Dr.
Roanoke, VA 24019
(703) 362-3960
Larry Rackow

Commodore Users of Franklin
1201 N. High St.
Franklin, VA 23851
(804) 562-6823
D. Bruce Powell

WASHINGTON

NW PET Users Group
2565 Dexter N. 3203
Seattle, WA 98109
Contact: Richard Ball

PET Users Group
c/o Kenneth Tong
1800 Taylor Ave. N102
Seattle, WA 98102

Whidbey Island Commodore
Computer Club
947 N. Burroughs Ave.
Oak Harbor, WA 98277
Michael D. Clark

Central Washington
Commodore Users Group
1222 S. 1st St.
Yakima, WA 98902
Tim McElroy

Blue Mountain Commodore
Users Club
15 Stone St.
Walla Walla, WA 99362
(509) 525-5452
Keith Rodue

Spokane Commodore User Group
N. 4311 Whitehouse
Spokane, WA 99205
(509) 328-1464
Stan White

CBM Users Group
803 Euclid Way
Centralia, WA 98531
(206) 736-4085
Rick Beaber

Computer Club
c/o Honeywell, Inc.
5303 Shilshole Ave., NW
Seattle, WA 98107
(206) 789-2000
John Goddard

WEST VIRGINIA

Personal Computer Club
P.O. Box 1301
Charleston, WV 25325
Cam Cravens

TriState Commodore Users
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

Commodore User Group
73 Pine Hill Estates
Kenova, WV 25530
(304) 453-2124
Marc Hutton

WISCONSIN

Sewpus
c/o Theodore J. Polozynski
P.O. Box 21851
Milwaukee, WI 53221

Waukesha Area Commodore
User Group (WACUG)
256½ W. Broadway

Waukesha, WI 53186
Contact: Walter Sadler
(414) 547-9391

Commodore 64 Software Exchange Group
P.O. Box 224
Oregon, WI 53575
E. J. Rosenberg

C.L.U.B. 84
6156 Douglas Ave.
Caledonia, WI 53108
(414) 835-4645 pm
Jack White
2nd Sat every month 10:00 am

VIC-20 & 64 User Group
522 West Bergen Dr.
Milwaukee, WI 53217
(414) 476-8125
Mr. Wachtl

Menomonie Area Commodore Users Group
510 12th St.
Menomonie, WI 54751
(715) 235-4987
Mike Williams

C.U.S.S.H.
3614 Sovereign Dr.
Racine, WI 53406
(414) 554-0156
Tim Tremmel
3rd Saturday of month

Madison Area Commodore Users Group
1552 Park St.
Middleton, WI 53562
(608) 831-4852
John Carvin
3rd Thurs. each month

S.W.I.T.C.H.
W156 N8834 Pilgrim Rd.
Menomonee Falls, WI 53051
(414) 255-7044
Len Lutz

Milwaukee Area CBM64 Enthusiasts (M.A.C.E.)
P.O. Box 340
Elm Grove, WI 53122
(414) 259-5991
Kevin Wilde

The Eau Claire CBM64 Users Group
Rt. 5, Box 179A
Eau Claire, WI 54703
(715) 874-5972
John Slavsky, Jr.
2nd Thurs. 7 p.m.

WAVE
P.O. Box 0641
Waukesha, WI 53187

WYOMING

Commodore Users Club
c/o Video Station
670 North 3rd #B
Laramie, WY 82070
(307) 721-5908
Pamela Nash

AUSTRALIA

WA VIC-Ups (VIC 20/CBM 64 Users)
14 Glengariff Dr.
Floreat Park 6014
Western Australia
B. J. Cook

AUSTRIA

Commodore Users Club
Postfach 5026
Salzburg
Austria
062-222-5391
D.A. Stagg

CANADA

Toronto PET Users Group, Inc.
1912A Avenue Rd., Ste. 1
Toronto, Ontario, Canada
M5M 4A1
(416) 782-8900
or call 416-782-9252
Contact: Chris Bennett

PET Users Club
c/o Mr. Brown
Valley Heights Secondary School
Box 159
Langton, Ont. N0E 1G0

Vancouver PET Users Group
P.O. Box 91164
West Vancouver, British Columbia
Canada V7V 3N6

CCCC (Canadian Commodore Computer Club)
c/o Strictly Commodore
47 Coachwood Place
Calgary, Alberta, Canada
T3H 1E1
Contact: Roger Olanson

W.P.U.G.
9-300 Enniskillen Ave.
Winnipeg, Manitoba R2V 0H9
Larry Neufeld

VIC-TIMS
2-830 Helena St.
Trail, British Columbia
V1R 3X2
(604) 368-9970
Greg Goss

Arva Hackers
Medway High School
Arva, Ontario N0M 1C0
D. Lerch

Nova Scotia Commodore Computer Users Group
66 Landrace Cres.
Dartmouth, N.S. B2W 2P9
Andrew Cornwall

Bonnyville VIC Cursors
Box 2100
Bonnyville, Alberta T0A 0L0
(403) 826-3992
Ed Wittchen

Commodore Users Club of Sudbury
938 Brookfield Ave.
Sudbury, Ontario
P3A 4K4

PET Educators Group
P.O. Box 454
Station A
Windsor, Ontario
N9A 6L7

COMVIC
P.O. Box 1688
St. Laurent
Montreal, Quebec
H4L 4Z2

Calgary Commodore Users Group
37 Castleridge Dr., N.E.
Calgary, Alberta
T3J 1P4
John Hazard

Fledging Barrie User Group (BUG)
58 Steel St.
Barrie, Ontario
Canada L4M 2E9

FINLAND

VIC-Club in Helsinki
c/o Matti Aarnio
Linnustajankj 2B7
SF-02940 ESP00 94
Finland

GERMANY

Kettenberg 24
D 5880 Lueden Scheid
West Germany
Rudi Ferrari

ITALY

Commodore 64 Club
Universita di Studi shan
V. Avigliana 13/1
10138 TORINO
ITALY

KOREA

Commodore Users Club
K.P.O. Box 1437
Seoul, Korea
Contact: S. K. Cha

MEXICO

Asociacion De Usarios Commodore
c/o Alejandro Lopez
Arechiga
Holbein 174-6° Piso
Mexico 18, D.F.

Club de Usarios Commodore
Sigma del Norte
Mol del Valle, Local 44
Garza Garcia, N.L. 66220

Club Microvic
Villaldama 225
Col. Chapultepec
Monterrey, N.L.
Mexico 66450
Oscar Sosa, President

NEW ZEALAND

Commodore Users Group
Meet at VHF Clubrooms
Hazel Ave.
Mount Roskill
3rd Wed. of month, 7:30 pm
Roger Altena 278-5262

Nelson VIC Users Group
c/o P.O. Box 860
Nelson, New Zealand
Peter Archer

E.R. Kennedy
c/o New Zealand Synthetic Fuels Corp. Ltd.
Private Bag
New Plymouth

NORWAY

VIC Club of Norway
Nedre Bankegt 10,
1750 Halden
Norway

UNITED KINGDOM

North London Hobby Computer Club
Dept. of Electronics & Communications Engineering
The Polytechnic of North London
Holloway Rd.
London N7 8DB

Croydon Microcomputer Club
111 Selhurst R.
Selhurst, London SE25 6LH
01-653-3207
Vernon Gifford

User Bulletin Board

Commodore Computer Camp

Registration for the second annual Commodore Computer Camp being held July 8-13, 1984, at Lincoln College in Lincoln, Illinois, is now underway. Participants will rub elbows with and bask in the knowledge of such noted Commodore wizards as Jim Butterfield, Dick Immers, Keith Peterson, Len Lindsay, Jim Strasma and Loren Wright. For brochure, write *immediately* to Director of Continuing Education, Lincoln College, Lincoln, Illinois 62656. Enrollment is limited.

TPUG Conference

The third annual Toronto PET User's Group conference will be held in Toronto May 26-27, 1984, at the Constellation Hotel, 900 Dixon Road. Full two-day program of speakers, exhibitors of hardware and software, ten-minute free consultation in "Answer Room" and Trader's Corner for used equipment. Pre-registration, including one year's Associate Membership, is $40.00. Contact Chris Bennett at TPUG (see User Group Listing).

An Introduction to Database Management with *The Manager*

by Thomas Ziegler
Commodore Software

The Manager ***is a full featured, powerful database manager for the Commodore 64 and SX-64 computers. It is a program that allows you to design your own applications and tailor them to your needs.***

The Manager can be used to collect and organize information in your home or business. In your home, it can help you organize your checkbook, your stamp collection, your Little League team, your investments, your Christmas card list or your recipes. In your business, it can keep track of inventory, personnel, accounts payable, sales or accounts receivable.

By providing the necessary tools to make information work for you, *The Manager* puts you in control.

What is Database Management?

Database management is a widely used, potentially confusing term, although it shouldn't be confusing because at one time or another all of us have had to manage a database. You probably didn't realize that you have been a database manager, but you have.

Your checkbook, for example, is a database. It contains data that describes the state of your checking account. This data is in a consistent format. For each check, you enter the date, the check number, amount of the check and to whom the check was written. All the entries in your checkbook look more or less the same. Similarity of entries is important when you are dealing with a computer database because the computer takes advantage of the similarity of the data to organize it.

Your address book is also a database. You keep, probably in a small black book, names and addresses of all your friends and relatives. Your address book has many different entries, but each entry has the same basic format: name, address and telephone number. If you kept your address book on a computer, your entries would be stored on a magnetic disk instead of in a book. On the computer, each entry is called a record. Your uncle Harry's name, address, and phone number, for instance, would be one record. Many records make a file. A database manager manages and organizes records and files.

The individual pieces of Uncle Harry's record (i.e., his first name, last name, address, city, state, zip code and telephone number) are called fields. Field, record and file are important terms to remember when you are learning about databases on a computer. Figure 1 shows the relationship of these items.

Let's take a look at the checkbook again. Each check is a record. Here are the fields common to every record:

Figure 1. Records, Fields, Files

Field 1	Field 2	Field 3	Field 4	
John Smith	100 Main Street	Anytown	PA	Record 1
Jane Doe	1525 Oak Lane	Springfield	NJ	Record 2
Bill Brown	900 Broadway	Anytown	PA	Record 3

File

Records are made up of fields. A file is a group of records.

1. Check number
2. Date
3. Payee (to whom the check is written)
4. Amount

To use a computer as a database manager you can either write your own program or use a professionally developed product like *The Manager.* When you use *The Manager,* you don't have to worry about random access, sequential files and the like. All you have to concern yourself with is your data. *The Manager* allows you to create a video fill-in-the-blanks form into which you enter your records. You can design the form to accommodate virtually any type of data you need to keep track of. After you enter data into the form, *The Manager* stores the data as a file on a diskette.

What can *The Manager* do with the database it manages? The easiest way to understand what a database manager can do is to see examples of databases and their uses. In the next few paragraphs, I will give you some examples of applications that could be implemented with *The Manager.*

Example 1: XYZ Computer Club Mailing List

The XYZ computer club sends a monthly newsletter to all its members and to a few people who aren't members but are interested in the club. The club uses *The Manager* to maintain its mailing list. Here are the fields kept for each record in the mailing list file:

1. Last name
2. First name
3. Address
4. City
5. State
6. Zip code
7. Telephone number
8. Membership status
9. Date member joined the club
10. Office held

The club uses this database to produce mailing labels sorted in zip code order for their newsletter.

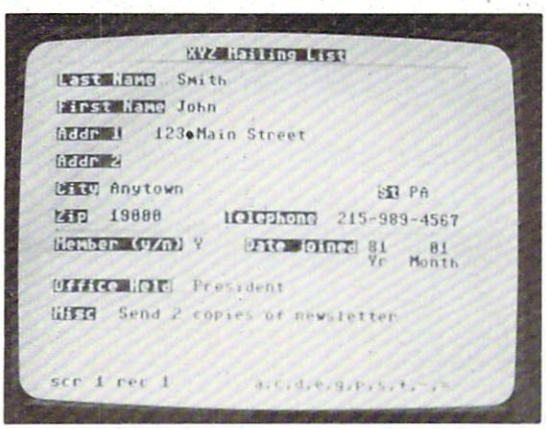
The Manager has the ability to alphabetically or numerically sort randomly entered data. Sorting data is one of the things a database manager is good at. To sort the mail by hand would take hours.

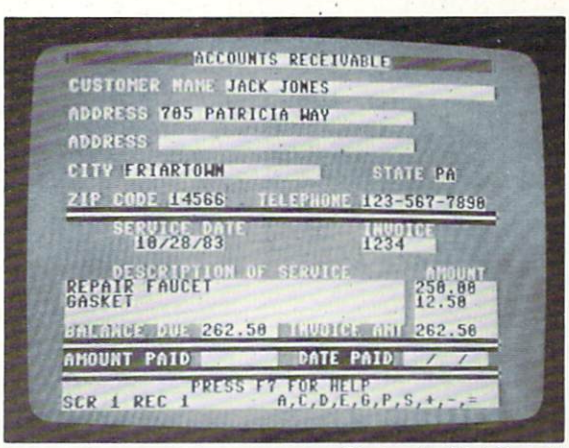
The club also uses *The Manager* with the *EasyScript* word processor from Commodore to produce personalized form letters whenever a special event is taking place. *The Manager* has the ability to create a file in a format that *EasyScript* can use for mail merge.

Example 2: Acme Plumbing Accounts Receivable

This company uses *The Manager* to keep track of their accounts receivable. The fields for each record are:

1. Invoice number
2. Date of service
3. Customer name
4. Customer address
5. Customer city
6. Customer state
7. Customer zip code
8. Customer phone number
9. Service rendered
10. Amount owed
11. Amount paid
12. Date paid





Photos by Mel Grauel

Acme Plumbing also uses *The Manager* to do their monthly billing. Before they had the computer, they had to type each bill. Now the computer does the typing and the sorting. With *The Manager* they can also get information that they never had before because it took too long. They can get a list of all unpaid bills older than 30 days, listed alphabetically by customer. They can get income reports on demand. And they can determine which kinds of jobs they do most often.

Example 3: Acme Plumbing Basketball Team

Acme Plumbing also sponsors a basketball team. The team wanted to keep detailed statistics on each player but no one had the time to do all the calculations. Now they use *The Manager* to keep statistics on each player. All they need to do is enter the scoresheets into the database after each game. Here are the fields they use:

1. Player's name
2. Date of game
3. Opponent
4. Shots taken
5. Shots made
6. Free throws attempted
7. Free throws made
8. Total points
9. Rebounds
10. Fouls
11. Assists

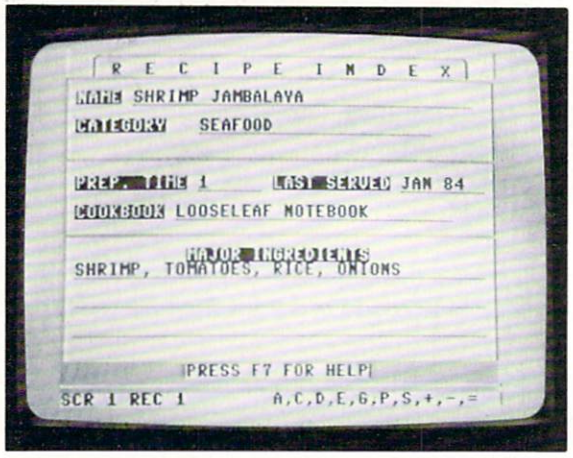
The Manager can calculate for every player on the team: shooting percentage, free throw percentage, average points per game, average rebounds per game and average assists per game. *The Manager* can also determine each player's averages against specific opponents.

Example 4: Judy Doe Recipe Index

Judy Doe loves to cook. She has at least 15 recipe books with hundreds of recipes from all over the world. Her problem was that she didn't always have the time to search through all her books for a good recipe. She decided to use *The Manager* to index her recipes. Here are the fields she uses:

1. Name of dish
2. Category (e.g., beef, dessert, vegetarian, etc.)
3. Ease of preparation
4. Preparation/cooking time
5. Date last served
6. Major ingredients
7. Name of cookbook and page number with recipe

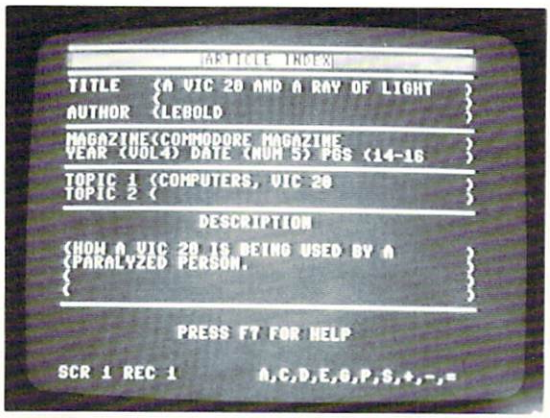
Now when Judy sees beef is on sale, she can quickly scan her database and find all her beef recipes, note when she last served a dish, how long it takes to prepare, what ingredients it uses and in which book the recipe is.



Example 5: John Smith Research Article Index

John Smith is a graduate student working on his masters degree. He does a lot of research and had trouble keeping track of all the articles he read. He used *The Manager* to set up the following database:

1. Name of article
2. Author
3. Journal/magazine
4. Date



5. Page numbers
6. Major topics covered
7. Brief description of the article

Now whenever John wants to find all the articles he's read on a specific subject, he can use *The Manager* to find them. It will give him a list by journal and date of all the articles he needs to review.

All of these examples of database management with *The Manager* have the following things in common:

1. *The Manager* provides information that was previously unavailable because manual methods were too time consuming.

2. It organizes the data in a consistent, easily accessible format.

3. It increases each person's productivity. Each of these people or groups can get more work done than they previously could.

4. It makes the data work *for* them, rather than against them.

5. None of these people is a programmer, yet each one designed an application tailored to his/her needs.

6. They all use *The Manager* as a tool to help them, not as a replacement for their own knowledge and experience.

A database manager is an electronic filing cabinet. But unlike the files in a regular filing cabinet, electronic files can be sorted, reorganized, manipulated and scanned at any time, very quickly.

The information in a regular filing cabinet is static. The filing cabinet stores the information but you can't use this information unless you are willing to sort through it and reorganize it. But the information in an electronic filing cabinet is dynamic. It is there to be used. Sorting and reorganizing the information are done at computer speed with little or no operator intervention.

To make it easy for you to get started with database management, Commodore is marketing 100 ready-to-use applications specifically designed for *The Manager.* Included in these applications will be home inventory, home budgeting, recipe indexing, sports statistics, business accounting, coupon inventory, wine list and many more. These applications will provide you with an introduction to the many capabilities and possibilities of database management with *The Manager.*

Table 1. Features Overview

- 1500 character records
- 2000 records per file
- 250 fields per record
- Sorting on 16 different keys
- 20 data entry screens per file
- Indexing on any field
- Report generator
- Arithmetic calculations
- Combination and/or searches

C

that does not compute...

"COMAL Graphics"
by Len Lindsay
Issue 27

The article was based on a preliminary release of the Commodore 64 version of COMAL. The following corrections are based on the final release.

Pages 88 & 89 should be: Enter graphics mode with the command SETGRAPHIC 0

Page 89 should be: To go into text mode, simply issue the command SETTEXT.

Page 90 should be: (to go to text screen): SETTEXT

Pages 91-93 (COMAL Graphics Keywords):

CLEARSCREEN should be CLEAR
Delete CLEARTEXT
DRAW should be SETGRAPHIC 0
Delete HIDESCREEN
NODRAW should be SETTEXT
Delete SHOWSCREEN

Also, the following are not implemented with disk-loaded COMAL, although they will be part of the COMAL cartridge for the 64:

CORRECTION, GETCOLOR, HEADING, SCALE, SETX, SETY, XCOR, YCOR.

Correct prices are: PET/CBM disk $15; Commodore 64 disk $19.95.

"COMAL Sprites"
by Len Lindsay
Issue 28

This article, too, was based on the preliminary release of COMAL for the Commodore 64. These corrections are based on the final release.

Line 0010 of the first program listing: should be 0010 SETGRAPHIC 0

To turn on a sprite: command is IDENTIFY

In the Sprite Chart:

Delete SHOWSPRITE
Fix SPRITEBACK to the following:
SPRITEBACK <color1>,<color2>
SPRITEBACK 1,0 sets the two common multicolors for sprites.

Also note: the 64th byte of a sprite definition string is not a check byte. If it is a CHR$(0) it means HI RES sprite. Any other number means MULTI-COLOR sprite.

Prices should be $15 for PET/CBM disk and $19.95 for Commodore 64 disk.

"New Products" page 122
Issue 26

MicroTie Systems Corporation in Walnut Creek, California, announced the Byte Bat in that issue. However, readers tell us MicroTie is not responding to orders and the toll-free number listed is no longer in service.

STUDENT SCHEDULING

SCHEDULE JUNIOR OR SENIOR HIGH

- OVERNIGHT TURNAROUND
- REDUCED EXPENSE

COMMODORE 8032 COMPUTER AND 8050 DISK DRIVE

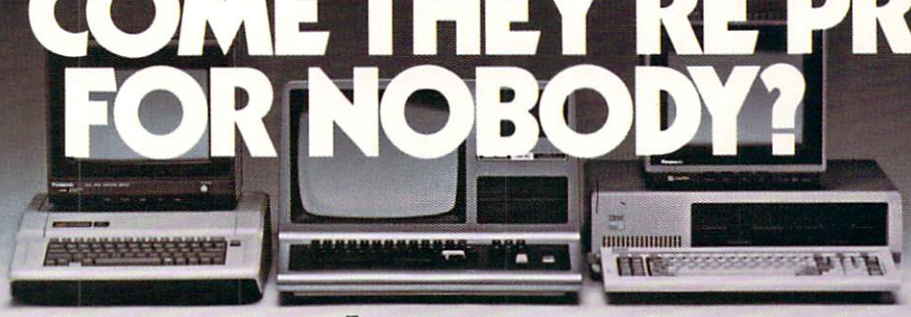
The Mainframe System That Runs on A Micro

COW BAY COMPUTING
Box 515
Manhasset, New York 11030
(516) 365-4423

advertisers' index

Advertisers	Page No.
Academy Software	106
C.S.I.	22
Cardinal Software	95
Cheatsheet Products	85
Commodore	IFC & 1, 7, 63
Continental Press	15
Covox Co.	28
Cow Bay Computing	128
Handic Software	11
Input Systems	16
Midwest Software	85
New Leaf	86
Peek Software	20
Public Domain	15
St. Croix Valley Electronics	103
Sota Enterprises	17
Star Micronics	5
SubLogic	9, 13
Suckle	86
Vin Systems	28

IF PERSONAL COMPUTERS ARE FOR EVERYBODY, HOW COME THEY'RE PRICED FOR NOBODY?



$1395* APPLE® IIe 64K

$999* TRS-80® III 16K

$1355* IBM® PC 64K

A personal computer is supposed to be a computer for persons. Not just wealthy persons. Or whiz-kid persons. Or privileged persons.

But person persons.

In other words, all the persons whom Apple, IBM, and Radio Shack seem to have forgotten about (including, most likely, you).

But that's okay. Because now you can get a high-powered home computer without taking out a second mortgage on your home.

It's the Commodore 64. We're not talking about a low-priced computer that can barely retain a phone number. We're talking about a memory of 64K. Which means it can perform tasks most other home computers can't. Including some of those that cost a lot more. (Take another look at the three computers above.)

By itself, the Commodore 64 is all the computer you'll ever need. Yet, if you do want to expand its capabilities some day, you can do so by adding a full complement of Commodore peripherals. Such as disk drives. Modems. And printers.

You can also play terrific games on the Commodore 64. Many of which will be far more challenging than those you could ever play on a game machine alone.

And as great as all this sounds, what's even greater-sounding is the price. It's hundreds of dollars less than that of our nearest competitor.

So while other companies are trying to take advantage of the computer revolution, it seems to us they're really taking advantage of something else:

Their customers.

*Manufacturers' suggested list prices. Monitor included with TRS-80 III only. Commodore Business Machines–P.O. Box 500R, Conshohocken, PA 19428; Canada–3370 Pharmacy Avenue, Agincourt, Ont., Can. M1W 2K4.

commodore COMPUTERS

Apple is a registered trademark of Apple Computer, Inc.
TRS-80 is a registered trademark of Tandy Corp. IBM is a registered trademark of International Business Machines Corp.

THE COMMODORE 64™ UNDER $300.*
You can't buy a better computer at twice the price.

REMEMBER.

ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

ISBN 0-88731-006-0